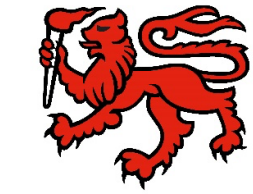


UNIVERSITY*of*
TASMANIA

Interactive Visualization for Data Inference in the Geosciences

by
Peter Edward Morse
Ph.D., B.A. (Hons), B.A.

Submitted in fulfilment of the requirements for the degree of
Doctor of Philosophy

Earth Sciences
School of Natural Sciences | College of Sciences and Engineering

March, 2021

Page Left Blank Intentionally

DECLARATION OF AUTHORSHIP

I, Peter E. MORSE, declare that this thesis titled, "Interactive Visualization for Data Inference in the Geosciences" and the work presented in it are my own.

Declaration of Originality. This thesis contains no material which has been accepted for a degree or diploma by the University or any other institution, except by way of background information and duly acknowledged in the thesis, and to the best of my knowledge and belief no material previously published or written by another person except where due acknowledgement is made in the text of the thesis, nor does the thesis contain any material that infringes copyright.

Authority of Access. The non-published content of the thesis (see below) may be made available for loan and limited copying and communication in accordance with the Copyright Act 1968.

Statement Regarding Published Work Contained in Thesis. Chapter 4 (Paper 1) is published by Elsevier under the CC BY-NC-ND license. Chapter 5 (Paper 2) of this thesis is published by Frontiers in Earth Science under a Creative Commons Attribution (CC BY) license. You are free to copy, communicate and adapt the work, so long as you attribute the authors.
To view a copy of this license, visit http://creativecommons.org/licenses/.
Chapter 6 (Paper 3) has been published by IEEE TVCG, and access to the material should be sought from the journal.

Peter Morse, 26th March, 2021

Statement of Co-Authorship

The following people and institutions contributed to the publication of work undertaken as part of this thesis:

Candidate - **Peter Edward Morse**, CODES, School of Natural Sciences (Earth Sciences), University of Tasmania.

Author 1 - Anya Marie Reading, School of Natural Sciences (Physics), University of Tasmania.

Author 2 - Tobias Stål, School of Natural Sciences (Earth Sciences) & Institute for Marine and Antarctic Studies, University of Tasmania.

Author 3 - Christopher Lueg, Berner Fachhochschule, Engineering and Information Technology, Abt Medizininformatik, Biel, Switzerland.

Author Details and Their Roles

Paper 1, located in Chapter 4.

Morse, P., Reading, A., Lueg, C., 2017. Animated analysis of geoscientific datasets: An interactive graphical application. Computers & Geosciences 109, 87–94. https://doi.org/10/gcmxvw

Paper 2, located in Chapter 5.

Morse, P.E., Reading, A.M., Stål, T., 2019. Well-Posed Geoscientific Visualization Through Interactive Color Mapping. Front. Earth Sci. 7, 274. https://doi.org/10/ggbjzq

Paper 3, , located in Chapter 6.

Morse, P.E., Reading, A.M., Stal, T., 2020. Exploratory volumetric deep Earth visualization by 2.5D interactive compositing. IEEE Trans. Visual. Comput. Graphics 1–1. https://doi.org/10/gh5n22

The candidate was the primary author on each paper.
Author 1 contributed to the case study. Author 1 and 3 contributed to the refinement and presentation of paper 1. Author 1 and 2 contributed to the case study, refinement and presentation of paper 2 and 3.

Signed:

Peter Morse Candidate School of Natural Sciences (Earth Sciences) University of Tasmania	Anya M. Reading Supervisor School of Natural Sciences (Physics) University of Tasmania	Sebastien Meffre Head of Discipline School of Natural Sciences (Earth Sciences) University of Tasmania

Date: 26th March 20201

"To-morrow, and to-morrow, and to-morrow,
Creeps in this petty pace from day to day,
To the last syllable of recorded time;
And all our yesterdays have lighted fools
The way to dusty death. Out, out, brief candle!
Life's but a walking shadow, a poor player
That struts and frets his hour upon the stage
And then is heard no more."

- Macbeth, William Shakespeare, 1623.

UNIVERSITY OF TASMANIA

ABSTRACT

Discipline of Earth Sciences
School of Natural Sciences

Doctor of Philosophy

Interactive Visualization for Data Inference in the Geosciences

by Peter E. MORSE

Visual displays are a formidable means of conveying information to the human brain. They facilitate the formation of scientific knowledge about the physical world, based on underlying observations of diverse kinds, through representations that are understood by practitioners of the relevant discipline area. Such data visualizations are critical in the geosciences given the need to draw meaning from time-varying, spatial or volumetric data, and given the increasing size of the datasets available for analysis of the natural, physical world.

The research described in this thesis aims to apply a novel set of technical resources to visualization in the geosciences. It draws on the immense potential of the human user for feature detection through connecting scientific data formats to computer graphics technologies. The software applications written in response to this opportunity therefore make strong use of interactivity in the reconnaissance exploration of example datasets. Throughout the research, a commitment to a well-posed visual display is developed, respecting underlying data values through the managed use of color and other graphic variables.

Following a review of the conceptual background, and the landscape of computer graphics technologies, the first original research chapter presents interactive software and workflows to visualize large geoscientific time-series datasets. It uses an animated interface and Human-Computer Interaction (HCI) to utilize the capacity of human expert observers to identify features via enhanced visual analytics. User-generated metadata allows subsets of the data to be tagged for subsequent closer investigation. The tool provides a rapid pre-pass process using fast GPU-based OpenGL graphics and data-handling. It makes use of interoperable data formats, and cloud-based (or local) data storage and computation. In a case study, the software was used to characterize a decade (2000–2009) of data recorded by the Cape Sorell Waverider Buoy, located approximately 10 km off the West coast of Tasmania, Australia. These data serve as a proxy for the understanding of Southern Ocean storminess, which has both local and global implications. Four different types of storm and non-storm events are characterized and compared with conventional analysis, noting the advantages and limitations of data analysis using animation and human interaction.

The second original research chapter presents a suite of newly written computer applications for 2D data, which enable spatially varying data to be displayed and analyzed in a performant graphics environment. Color-mappings using illustrative color spaces (RGB, CIELAB) are compared with the aid of interactive displays of the applied gradient paths through the chosen color spaces. This facilitates the creation of color-maps that accommodate the non-uniformity of human color perception, producing an image where genuine features are seen, taking account of aspects of the data such as parameter uncertainty. For an illustrative case study using a seismic tomography result, interpolation in CIELAB color space is shown to enable the creation of perceptually uniform linear gradients that match the underlying data, along with a simply computable metric for color difference, ΔE. This color space assists the accuracy and reproducibility of visualization results.

The well-posed use of color is further developed in the third original research chapter, for the exploratory interactive visualization of 3D volumes of global, deep Earth data. As an example, we address the challenge of reconnaissance visualization of a combined seismic tomography result, the primary means by which geoscientists infer structure and process in the deep Earth. A novel, interactive graphical application suite is presented that uses an intuitive 2.5D layer compositing approach. This allows the user to adjust the separation between data-slices, control graphics variables such as color mapping, opacity and compositing, and enables exploration and annotation of the architecture of the lithosphere. The methodology could find use in the visualization of multiple datasets representing aspects of the Earth's deep interior, oceans and atmosphere, and in facilitating researcher interaction with the increasing number of rich datasets from missions to our neighboring planets.

The three original research papers that form the core of this thesis all provide a means of amplifying analytical acuity through animated and/or interactive interfaces that enable both 'overview' and 'detail' visualization and navigation. Through all three studies, the 'human in the loop' aspects of the visualization process are drawn upon, e.g. in the use of perceptual color spaces for optimal display of data, or exploiting visual faculties such as stereopsis and depth perception.

The dataflow software methodology employed is self-documenting, using a visual programming approach that can be replicated in alternative cross-platform software environments such as recent computer game engines. This flexible strategy assists the development of novel graphical user interfaces and interaction modalities for collaborative immersive screen technologies such as domes and future XR applications.

In summary the research described herein bridges the gap between scientific data formats and the immense resources of the computer graphics and gaming industries. It exploits productive modes of HCI engagement with the data display to facilitate the search for new knowledge in the geosciences. It is anticipated that the newly written software applications will lead to wider usage of informed color-mapping in the geosciences and an awareness of the utility of emergent visualization platforms for enhancing scientific research. It is hoped that "visual literacy" and "visual numeracy" will substantially improve as a consequence of this work, and similar initiatives, as inference tasks are more routinely carried out using well-posed data visualization in the geosciences.

Table of Contents

List of Figures

List of Tables

ACKNOWLEDGEMENTS

This PhD research was supported by an RTP University of Tasmania Scholarship.

I wish to extend my thanks to Anya Reading for her generosity, knowledge and resilience. To my friend Tobias for collaboration. To Chris Lueg for support at the outset, to Steve Walters for constructive feedback. To Jodi, Dan and Martin for laughter. Thanks also to my other colleagues in the Compute Earth Group and CODES.

Most especially my thanks to Team EEG, for their amazing patience and love. And my parents, who have lived to witness the ending of yet another madcap adventure.

Any mistakes in the text are my own.

It's been fun.

Peter Morse
Gordon, Tasmania
September 2020

Chapter 1 : Introduction

1 Thesis Motivation

Scientific visualization applied to the visual analysis of data can benefit deeply from the immense technical resources, technological developments, design and programming practices enabled by creative industries technologies (CIT), such as interactive media, computer game engines and digital cinema. Modern computer graphics and modes of human-computer interaction provide a wealth of approaches and insights that are both qualitative and quantitative in nature. Scientific visualization should exploit this extensive knowledge domain to become more effective as a complement to conventional analytic methods.

Accurate representations in quantitative, spatial sciences, such as the geosciences, require visualization approaches that retain the rigor of numerical input whilst facilitating efficient workflows. This is particularly pressing given the massive increase in the volume and variety of data, and the need for an acceleration in the interpretation of spatial data. We can make use of the practical convergence between these goals and the technical capabilities afforded by performant Graphics Processing Units (GPUs). Accelerated graphics and interaction capabilities can be combined with cloud-enabled ability to connect with scientific data formats. Concurrent with this opportunity is the need to increase 'graphical literacy' and 'graphical numeracy' for geosciences, including an understanding of the intent of visualization and its efficacy. Visualization exhibits great strengths in its capacity to engage, inform, explore, and facilitate collaboration in research.

To realize these objectives, this thesis is directed towards well-posed visualization for geoscientific data exploration and inference, supported by newly written software and novel analysis workflows, for interactive inquiry and knowledge generation.

1.1 Research Aims

This research has five main aims:

1. Exploit computer graphics technologies and modes of Human-Computer

Interaction (HCI) from CIT to enable the exploration for new knowledge in the geosciences.

2. Amplify analytical acuity through animated and/or interactive interfaces that enable both 'overview' and 'detail' visualization and navigation.
3. Account for 'human in the loop' visualization, considering aspects of human perception in an iterative visualization process, e.g. in the use of perceptual color spaces for optimal display of data, or exploiting visual faculties such as stereopsis and depth perception.
4. Develop methodologies and an enriched toolset that facilitates well-posed geoscientific visualization, including the ability to interface with scientific data formats, to respect underlying data values, and to characterize uncertainty.
5. Anticipate the evolution of geoscientific visualization software beyond conventional Windows-Icons-Menus-Pointers (WIMP) approaches, exploring prospective applications for shared, immersive and large-scale screen technologies for collaborative visualization.

Underpinning these aims are two inter-related research imperatives. Firstly, how can animated, interactive visualization enhance scientific inference in the geosciences? Secondly, which aspects of CIT approaches are useful in pursuit of these goals?

1.2 Visualization, Visual Analytics and Data Inference

Visualization can be defined as the communication of concepts and information using graphical representations (Ward et al., 2010, p. 1; Ware, 2013, p. 2). Computer-based visualization systems provide visual representations of datasets designed to help people carry out tasks more effectively (Munzner, 2014, p. 1). This research builds upon these definitions, incorporating implications for interaction, optimization and interpretation: visualization as an inferential activity, not only an end-product of research.

Static graphs and other diagrammatic representations of data can be thought of as visualizations (Few, 2009; Tufte, 1997, 1990). Existing visualization tools are frequently used for the presentation of these results to engage and inform, rather than forming an inherent part of inference in the sciences (Few, 2015; Victor, 2005; Ware, 2013). However, modern performant computers and high-performance graphics, interactive software and hardware implementations enable human-computer interactions during data visualization and analysis. The incredible potential for data exploration this opens up cannot be understated, with particular emphasis upon novel abilities to explore accurate interactive visual inference in systematic, analytical ways. Interactive, animated visualization affords methodological advantages for synoptic overview and visual reconnaissance of large datasets, with demonstrable advantages for visual analytics (VA)

and associated processes of scientific inference.

Visual analytics, a relatively recent field of research, has been defined as "the science of analytical reasoning facilitated by visual interactive interfaces" (Thomas and Cook, 2005). One key implication of visual analytics is that data visualization is not an end-product of scientific research, but an essential tool before, during and after research processes. Indeed, analytical visualization should be used much more widely at early stages in a scientific workflow, the process of visual data inference prospectively generating new insight into the underlying data (Keim et al., 2008; Thomas and Cook, 2005; Ward et al., 2010).

Data inference implies the process of transforming raw empirical data into information (data with defined constraints, causes and effects) and from this deriving scientific knowledge about the operations of the natural world (Floridi, 2011; Purchase et al., 2008). As an example, in the context of visual analytics (Keim et al., 2010), data undergoes transformation and filtering via statistical parsing (data mining), to form the basis of a model (data with constraints); in parallel, mappings are performed for visualization and human interlocution, feeding back into refined parameter constraints and model approximations. These iterative processes articulate human perceptual-cognitive systems in the development of inference within a given scientific epistemology (Popper, 1959). The considerable body of work on innovative quantitative inference in the geosciences is recognized (e.g. Sambridge et al., 2013), while this research explores approaches that are intended to be complementary.

1.3 Human Factors

We start from the premise that computers do not understand data but that humans do, and that humans excel at pattern recognition. Recognizing patterns in scientific data is key to scientific inference and forming knowledge about data. Computer-based analysis augments our abilities to detect patterns, or more generally, features in data. Visualization is suitable when there is a need to enhance human capabilities rather than replace people with computational decision-making methods (Munzner, 2014). Visual displays provide the highest bandwidth channel from the computer to the human (Ware, 2013, p. 2).

The human comprehension of graphical representations is complex, as it involves the interplay between physiology, visual perception, cognition and experience, together with factors such as information density, dataset size, graphic design and color (Healey and Enns, 2012; Ware, 2013). Some flexibility in a visualization can accommodate physiological and experiential differences in scientific data analysts. Expert human analysts can identify patterns, features or structures that are challenging for statistical classification systems. Both comprehension and pattern detection are influenced by the graphical abstraction of data and the levels of detail present (Carpenter and Shah, 1998; Friel et al., 2001) as well characteristics of the design idiom and system architecture via which a visualization is

articulated (Munzner, 2014). Scientific visualization must account for human factors as well as intrinsic characteristics of data such as its type, scale and complexity.

1.4 Visualizing Large and Complex Data

The volume of data generated by scientific instruments, sensor systems and computational models is growing at an increasing pace (Hey et al., 2009) and most geosciences disciplines and technologies exhibit this trend (Sellars et al., 2013). Typical datasets might comprise time variant observations at fixed geographic locations such as tide-gauges and oceanographic buoy data, spatially distributed data such as soil geochemistry analyses, or combined spatial/time variant data such as those observed by satellites and seismic surveys. Other large datasets include outputs of model simulations and forecasts. Interpretations of geosciences data are commonly carried out using graphs and maps, however, there is a necessary limitation on how much information can be presented in a single representation and how many graphs or maps can be usefully incorporated in one study (Munzner, 2014; Ware, 2013). Statistical analyses and machine learning approaches afford the ability to summarize in a systematic way but at the expense of exploratory analysis and much pattern characterization (Hammer et al., 2014; Keim et al., 2010; Tukey, 1977). Incorporating the human expert observer in the data inference process itself may prospectively lead to more rapid identification of observations and patterns of scientific significance, as well as recursively lead to improvements in automated machine learning and statistical techniques (Amershi et al., 2014; Bigdely-Shamlo et al., 2008; Cakmak and Lopes, 2012; Fiebrink et al., 2011; Pohlmeyer et al., 2011). Again, this suggests a feedback loop between observer and observation: an iterative process not only of analysis but also of research design, implementation and visual model-building, leading to reproducible results and accounting for uncertainty (Milton and Possolo, 2020).

1.5 Thesis structure

This thesis comprises three major parts:

Part One includes Chapters 1-3, being this introduction, followed by two chapters reviewing the relevant scientific visualization literature and technical background.

Part Two contains three chapters published/submitted to peer-reviewed journals as stand-alone manuscripts (some elements of which necessarily recapitulate preceding material).

Part Three contains the concluding Chapters 7 and 8. Chapter 7 synthesizes and discusses key results from the core chapters, including limitations of the approaches undertaken and suggests future work. Chapter 8 provides a brief summary of the research, the main findings and benefits for the geosciences.

1.6 Part 1: Introductory Materials

Chapter 1 – Introduction.

Introduces the opportunities and challenges for visual data analysis for the geosciences. Principal to this is the hypothesis that interactive computer visualization can enhance the scientific inference process, utilizing the human perceptual-cognitive system in exploratory comprehension of complex data. These interlocutions form a basis for sifting through data sets, identifying salient features for further expert analysis, and prospective statistical or other analytical techniques.

Chapter 2 – Conceptual Background.

A review of the challenges and contemporary scientific visualization solutions pertinent to geoscientific data visualization and data inference. This chapter provides context for the research, considering aspects of human perception and HCI. It develops a visualization workflow that aims to transform raw data into a well-posed visual form that can be explored in a way that enables scientific inference and induction.

Chapter 3 – Technical Review.

A review of computational tools and methods from scientific visualization, data visualization, information visualization and visual analytics and their intersections with CIT. It develops a visualization pipeline that identifies key technological and design approaches for their utility to future development.

1.7 Part 2: Published Research

Chapter 4. Enhanced Interactive Analysis of Time-series Data (Morse et al., 2017).

This chapter presents an innovative, interactive software tool and workflow to visualize, characterize, sample and tag large geoscientific time-series datasets from both local and cloud-based repositories. It uses an animated interface and HCI to utilize the capacity of human expert observers to identify features via enhanced visual analytics. The software enables users to analyze datasets that are too large in volume to be drawn legibly on a reasonable number of single static plots. Users interact with the moving graphical display, tagging data ranges of interest for subsequent attention. The tool provides a rapid pre-pass process using fast GPU-based OpenGL graphics and data-handling. It makes use of interoperable data formats, and cloud-based (or local) data storage and compute. In a case study, the software was used to characterize a decade (2000–2009) of data recorded by the Cape Sorell Waverider Buoy, located approximately 10 km off the west coast of Tasmania, Australia. These data serve as a proxy for the understanding of Southern Ocean

storminess, which has both local and global implications. This example shows use of the tool to identify and characterize 4 different types of storm and non-storm events during this time. Events characterized in this way are compared with conventional analysis, noting advantages and limitations of data analysis using animation and human interaction.

Chapter 5. Well-Posed Interactive Color-Mapping for 2D Seismic Data (Morse et al., 2019).

In this chapter we make use of a suite of newly written computer applications which enable spatially varying data to be displayed in a performant graphics environment. We present a comparison of color-mapping using illustrative color spaces (RGB, CIELAB). The interactive applications display the gradient paths through the chosen color spaces. This facilitates the creation of color-maps that accommodate the non-uniformity of human color perception, producing an image where genuine features are seen. We also take account of aspects of a dataset such as parameter uncertainty. For an illustrative case study using a seismic tomography result, we find that the use of RGB color-mapping can introduce non-linearities in the visualization, potentially leading to incorrect inference. Interpolation in CIELAB color space enables the creation of perceptually uniform linear gradients that match the underlying data, along with a simply computable metric for color difference, ΔE. This color space assists accuracy and reproducibility of visualization results. Well-posed scientific visualization requires both "visual literacy" and "visual numeracy" on an equal footing with clearly written text. It is anticipated that this current work, with the included color-maps and software, will lead to wider usage of informed color-mapping in the geosciences.

Chapter 6. Illustrative volumetric deep Earth visualization (Morse et al., 2020).

Building on the insights from the previous chapter, we extend our approach to 3D data volumes for the visualization of global, deep Earth volume datasets for display and researcher interaction. While the algorithms and data analysis techniques that produce such volumetric results have become more sophisticated, the manner of visualizing these findings can be improved. We address the challenge of making an illustrative, exploratory visualization of a global geoscience dataset using a combined seismic tomography result, the primary means by which geoscientists infer structure and process in the deep Earth. We present a novel, interactive graphical application suite and associated workflow that uses an intuitive 2.5D layer compositing approach. This allows the user to adjust the separation between data-slices, control graphics variables such as color mapping, opacity and compositing, and facilitate exploration and annotation of the architecture of the lithosphere. Graphics outputs from our applications are enabled for immersive display systems such as Fulldome and XR. In a case study we visualize deep Earth structure beneath the Indian Ocean region. We anticipate that the application methodology will find

use in the visualization of multiple datasets representing aspects of the Earth's deep interior and atmosphere, and in the interaction with the increasing number of rich datasets from missions to our neighboring planets.

1.8 Part 3: Synthesis, Discussion and Conclusion

Chapter 7 – Synthesis and Discussion. This chapter addresses how the principal aims of this research were achieved and consolidates the key findings. The visualization workflows, methodologies and software developed are discussed for their strengths and weaknesses, including limitations of the approaches undertaken. Future software development strategies are discussed, including capabilities of the software to interface with advanced display environments, including future potential for extending data exploration to collaborative contexts.

Chapter 8 – Conclusion. A brief summary of the main findings and demonstrated benefits for the geosciences.

1.9 References

Amershi, S., Cakmak, M., Knox, W.B., Kulesza, T., 2014. Power to the People: The Role of Humans in Interactive Machine Learning. AI Magazine 35, 105. https://doi.org/10.1609/aimag.v35i4.2513

Bigdely-Shamlo, N., Vankov, A., Ramirez, R.R., Makeig, S., 2008. Brain Activity-Based Image Classification from Rapid Serial Visual Presentation. IEEE Transactions on Neural Systems and Rehabilitation Engineering 16, 432–441. https://doi.org/10.1109/TNSRE.2008.2003381

Cakmak, M., Lopes, M., 2012. Algorithmic and human teaching of sequential decision tasks, in: AAAI Conference on Artificial Intelligence (AAAI-12).

Carpenter, P.A., Shah, P., 1998. A model of the perceptual and conceptual processes in graph comprehension. Journal of Experimental Psychology: Applied 4, 75. https://doi.org/10.1037/1076-898X.4.2.75

Few, S., 2015. Signal: Understanding what Matters in a World of Noise. Analytics Press.

Few, S., 2009. Now you see it: simple visualization techniques for quantitative analysis. Analytics Press.

Fiebrink, R., Cook, P.R., Trueman, D., 2011. Human model evaluation in interactive supervised learning, in: Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems - CHI '11. Presented at the 2011 annual conference, ACM Press, Vancouver, BC, Canada, p. 147. https://doi.org/10.1145/1978942.1978965

Floridi, L., 2011. The philosophy of information. Oxford University Press.

Friel, S.N., Curcio, F.R., Bright, G.W., 2001. Making Sense of Graphs: Critical Factors Influencing Comprehension and Instructional Implications. Journal for Research in Mathematics Education 32, 124. https://doi.org/10.2307/749671

Hammer, B., He, H., Martinetz, T., 2014. Learning and modelling big data, in: ESANN 2014 Proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Presented at the ESANN 2014, i6doc.com, Bruges, Belgium, p. 10.

Healey, C.G., Enns, J.T., 2012. Attention and visual memory in visualization and computer graphics. Visualization and Computer Graphics, IEEE Transactions on 18, 1170–1188. https://doi.org/10.1109/TVCG.2011.127

Hey, T., Tansley, S., Tolle, K.M., et al., 2009. The fourth paradigm: data-intensive scientific discovery. Microsoft research Redmond, WA.

Keim, D., Andrienko, G., Fekete, J.-D., Görg, C., Kohlhammer, J., Melançon, G., 2008. Visual Analytics: Definition, Process, and Challenges, in: Kerren, A., Stasko, J.T., Fekete, J.-D., North, C. (Eds.), Information Visualization. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 154–175. https://doi.org/10.1007/978-3-540-70956-5_7

Keim, D., Kohlhammer, J., Ellis, G., Mansmann, F., 2010. Mastering the information age solving problems with visual analytics. Eurographics Association.

Milton, M.J.T., Possolo, A., 2020. Trustworthy data underpin reproducible research. Nature Physics 16, 117–119. https://doi.org/10/ggktsr

Morse, P., Reading, A., Lueg, C., 2017. Animated analysis of geoscientific datasets: An interactive graphical application. Computers & Geosciences 109, 87–94. https://doi.org/10/gcmxvw

Morse, P.E., Reading, A.M., Stål, T., 2020. Resubmitted Following Review: Illustrative volumetric deep Earth visualization by 2.5D interactive compositing. IEEE Transactions on Visualization and Computer Graphics 14 pages.

Morse, P.E., Reading, A.M., Stål, T., 2019. Well-Posed Geoscientific Visualization Through Interactive Color Mapping. Front. Earth Sci. 7, 274. https://doi.org/10/ggbjzq

Munzner, T., 2014. Visualization Analysis and Design. A K Peters/CRC Press. https://doi.org/10.1201/b17511

Pohlmeyer, E.A., Wang, J., Jangraw, D.C., Lou, B., Chang, S.-F., Sajda, P., 2011. Closing the loop in cortically-coupled computer vision: a brain–computer interface for searching image databases. Journal of Neural Engineering 8, 036025. https://doi.org/10.1088/1741-2560/8/3/036025

Popper, K., 1959. The Logic of Scientific Discovery. Routledge.

Purchase, H.C., Andrienko, N., Jankun-Kelly, T.J., Ward, M., 2008. Theoretical Foundations of Information Visualization, in: Kerren, A., Stasko, J.T., Fekete, J.-D., North, C. (Eds.), Information Visualization. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 46–64. https://doi.org/10.1007/978-3-540-70956-5_3

Sambridge, M., Bodin, T., Gallagher, K., Tkalčić, H., 2013. Transdimensional inference in the geosciences. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 371, 20110547. https://doi.org/10/gg9vvd

Sellars, S., Nguyen, P., Chu, W., Gao, X., Hsu, K., Sorooshian, S., 2013. Computational Earth Science: Big Data Transformed into Insight. Eos, Transactions American Geophysical Union 94, 277–278. https://doi.org/10.1002/2013EO320001

Thomas, J.J., Cook, K.A., 2005. Illuminating the path: The research and development agenda for visual analytics. IEEE Computer Society Press.

Tufte, E.R., 1997. Visual explanations: images and quantities, evidence and narrative. Graphics Press, Cheshire, CT.

Tufte, E.R., 1990. Envisioning information. Graphics Press, Cheshire, CT, USA.

Tukey, J.W., 1977. Exploratory data analysis. Reading, Mass.

Victor, B., 2005. Magic Ink: Information Software and the Graphical Interface [WWW Document]. URL http://worrydream.com/MagicInk/ (accessed 2.14.20).

Ward, M.O., Grinstein, G., Keim, D., 2010. Interactive data visualization: foundations, techniques, and applications. AK Peters/CRC Press, Natick, Massachusetts.

Ware, C., 2013. Information visualization: perception for design, 3rd ed. Elsevier, Amsterdam, Holland.

Chapter 2: Conceptual Background

2 Introduction

Geoscientific visualization directly informs the process of scientific inference for the creation of new knowledge about the Earth and other planets. Data visualizations range from simple diagrams and conventional graphs drawn from a repertoire of graph types, such as histograms, bar charts, line plots, scatter plots and networks (Tufte, 1983; Wang and Tao, 2017; Wilkinson, 2005), to more complex representations that utilize 3D computer graphics technologies and advances in software and hardware capabilities (Hughes et al., 2013; Telea, 2015). As advances in computational machinery develop, so too do the capabilities and possibilities of geoscientific visualization, exposing new directions for innovation and improvement for the display and analysis of complex data.

Computer-based visualization

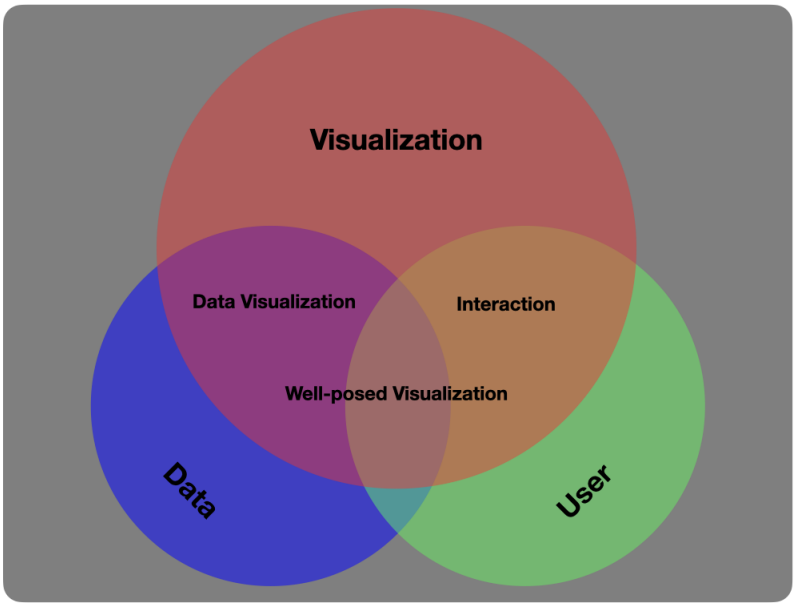


Fig. 1: Domain relationships relevant to interactive scientific visualization

Computer-based visualization is a mature scientific field (Reina et al., 2020), arising at the intersection of a wide array of disciplinary approaches and considerations (Keim et al., 2008). Following Van Wijk (2005) and Liu (2014), Fig. 1 identifies three key components of

a visualization activity: data, user and visualization. At their intersection is located a focal concern of this research: how 'well-posed' geoscientific visualizations can best be formed through data visualization and human-computer interaction.

Visual displays provide the highest bandwidth channel from computer to the human brain, using structured images to convey significant information about underlying data and processes (Hansen and Johnson, 2005). The motivation behind scientific visualization is to facilitate the formation of scientific knowledge about the physical world, utilizing visual representations that have coherent empirical and semiotic relationships with their underlying data sources (MacEachren, 2001; Purchase et al., 2008; Tufte, 1997). Key texts on data visualization address a broad range of disciplines including the psychophysics of perception (Ware, 2013), human-computer interaction (Ward et al., 2010), and analytics, graphical design and programming considerations (Hansen and Johnson, 2005; Munzner, 2014; Telea, 2015; Tukey, 1990; Wilkinson, 2005).

Ware (2013) addresses visualization from the perspective of 'visual thinking', based upon a wide-ranging study of the psychology and physiology of perception. It offers a comprehensive study of perception and first principles for visualization practice and interface design. Munzner (2014) builds upon this perspective, with a practical emphasis upon design implementation (e.g. abstract and idiomatic examples of design), focused by principles of task-orientation: what a visualization is meant to achieve and how this can be done. It provides many examples of implementation from both a designer and end-user's point of view, within a computer science framework. Ward et al. (2010) address data visualization from the perspective of algorithmic and software engineering issues, providing an introduction to programming for a wide variety of interactive visualization scenarios. Hansen and Johnson (2005), Telea (2015) and Wilkinson (2005) provide advanced graphics and statistical perspectives, covering technical implementation topics for software programming in, for instance, OpenGL, volumetric rendering and statistical graphics.

Other notable texts include Chen (2020), Few (2009, 2015a) and Tufte (Tufte, 1997, 1990, 1983), which provide critical analyses of the various pitfalls and best-practices of a wide range of visualization implementation and design strategies, with particular reference to effective and lucid representation of information for visual display.

Visualization workflow

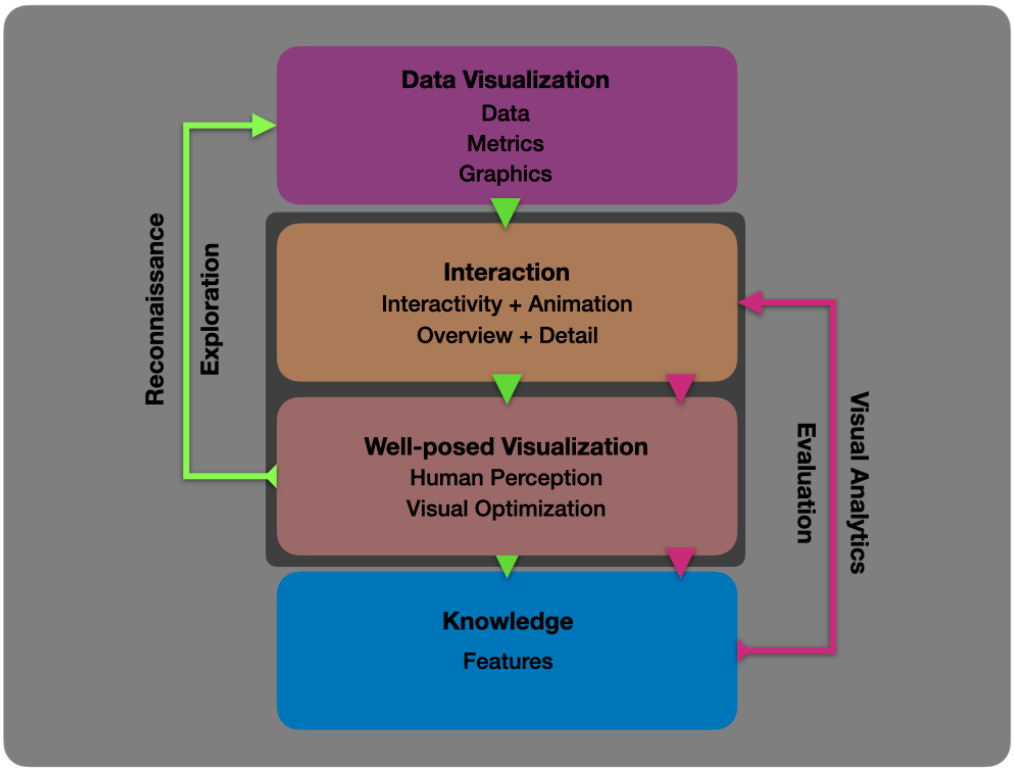


Fig. 2: Visualization Workflow

Rather than thinking of scientific visualization as an end product that illustrates, annotates or otherwise visually represents finished research products, it can be fully integrated into the process of scientific enquiry itself (Fox and Hendler, 2011). Fig. 2 represents the process of visualization as a workflow (Gil et al., 2007) that aims to transform raw data into a well-posed visual form that can be explored in a way that enables scientific inference and induction (Pearl, 2009; Popper, 1959).

The purpose of the workflow is to create accurate and reproducible visualizations that not only illustrate a scientific analysis, but that also actively assist in the process of scientific inference. The workflow divides the process of visualization into four stages with increasing levels of abstraction, connected by two iterative cycles. The two central stages are the focus of this research. The rest of this chapter is structured according to this workflow model, introducing relevant research and literature in each section, maintaining focus on practical approaches to visualization for knowledge discovery:

2.1 Data visualization. The first stage in the workflow concerns the basic transformation of data into visual representations. This section considers a taxonomy of scientific data, its metrics and graphical depiction.

2.2 Interaction. The second stage employs interactivity and animation for the purposes of 'overview and detail' navigation of a visualization, assisting the search for structure in data.

2.3 Well-posed visualization. The third stage considers human perception and how visual representations can be optimized for computer display and interaction. This includes the 'reconnaissance-exploration' loop, wherein the first three stages are iterated, providing a qualitative overview of the data 'terrain' to explore.

2.4 Knowledge. The final stage in the workflow considers derived knowledge about data and the ability to characterize its features. This includes the 'visual analytics – evaluation' loop, exposing both the means of visualization and the visualization itself to evaluative criteria.

2.1 Data Visualization

Research into data visualization has spawned an extensive and diverse literature (Kehrer and Hauser, 2013), with an equally diverse lexicon deriving from the research specialisms from which it derives: terminologies often exhibit different meanings within as well as across different disciplines (Craft and Cairns, 2008; Lau and Vande Moere, 2007). This can lead to some ambiguity, e.g. in the confusion/interchangeability of the terms 'data' and 'information', their nominal forms such as 'data visualization' and 'information visualization' and their portmanteaus: 'dataviz', 'infoviz', 'sciviz' etc.

It is worth noting that sometimes a distinction is drawn between 'data' and 'information', regarding data as raw collected measures, facts, or observations, whereas in computer-based information systems, 'information' includes both data and metadata that label the data. Metadata can include descriptions of the data, how they were collected, and numerous other features (Gray et al., 2005). For the purposes of this research, we use the terms 'data' and 'information' interchangeably, irrespective of data content type.

Data

The amount of data generated by instruments, sensor systems and computational models is growing at an exponential rate (Hey et al., 2009; Sellars et al., 2013). Notable data-intensive geoscientific disciplines and technologies include earth observation systems, seismic sensor data, satellite data, oceanographic buoy data, tectonic drift data, moored buoys, floats, geochemical analyses, model simulations and forecasts, amongst many others (Sellars et al., 2013). Data stored and managed in databases can be subjected to computational analysis (Chen et al., 1996; Fayyad, 2001; Gray et al., 2005), by fitting models to or determining patterns from observed data (Fayyad et al., 1996a). Data mining algorithms can perform functions upon data such as classification, regression, clustering,

summarization, dependency modelling, link analysis and sequence analysis (Fayyad et al., 1996b; Fayyad and Smyth, 1999). Second-order, preprocessed or derived data can form a significant component in the visualization workflow, especially if it is converted into machine-readable formats suitable for downstream visual processing (see Chapter 3).

Ward et al. (2010, p. 46) state that 'each observation or variable of a data record represents a single piece of information', subsequently categorizing these into Ordinal (binary, discrete, continuous) and Nominal (categorical, ranked, arbitrary) and additionally via Scale (ordering relation, distance metric, absolute zero value). Munzner (2014, p. 24) defines a dataset as 'any collection of information that is the target of analysis', providing a taxonomy of data, dataset types and cognate representations indicated in Table 1.

Dataset Types	Tables	Networks/Graphs & Trees	Fields	Geometry	Clusters, Sets, Lists
	Discrete	Discrete	Continuous/ Sampled/ Interpolated		
Data Types	Items (value)	Items (nodes/vertices)	Grids	Items (value)	Items (value)
	Attributes (value)	Links	Positions (Spatial)	Positions	Groups
		Attributes (value)	Attributes (value)		Categories
Data Structures + Dimensionality	2D-HD	1D-HD	2D-HD	2D-3D-HD	1D-HD
	Columns	Links (edge)	Grids	Spatial	Ordered
	Rows	Nodes	Cells	Coordinates	Unordered
	Cells	Trees	Columns		Semantic
	Keys		Rows		Keys
Data Structure Type					
	Flat		Scalar		Sequential
	Multidimensional		Vector		Diverging
			Tensor		Cyclic
					Topological
Temporal					
	Various	Various	Various	Various	Various

Table 1: Taxonomy of Dataset Types & Data (after Munzner, 2014)

Metrics

Data has types (e.g. floats, ints, categories etc.) and dimensionality (e.g. 1D, 2D, 3D, High-Dimensionality (HD), time variance). Metrics indicate suitable measures and scales for data, and parameterize how these can be algorithmically transformed into visual representations.

An important feature of data is its dimensionality, shown in Table 1. Simple 1D and 2D

datasets can be easily represented on 2-dimensional screen surfaces, 3D data (including volumetric and animated time variant data) can be represented in Cartesian coordinates via perspective and similar projections (e.g. dimetric, orthographic) as well as alternatives such as spherical coordinates and map projections (e.g. equirectangular, stereographic etc. (Snyder, 1987)).

Similarly, 4D, HD and other multivariate data can be depicted by, for example, complex plane projections or other exotic topologies (Carlsson, 2009), data slices through HD spaces or through parallel coordinates approaches (Inselberg, 2009, 2006). Dimensionality may not be spatial in origin or best expressed spatially in a visualization. Multivariate data analysis addresses datasets that may have large numbers of non-spatial dimensions – a common characteristic of geoscientific data (Cunningham et al., 2010; He et al., 2019; Wong and Bergeron, 1994). These HD data spaces present enormous challenges to 2D-screen based representation as well as the human visual-cognitive system (Chang et al., 2018). Representative studies present sophisticated techniques of dimensional reduction, data-binning and transformation by which high dimensional features can be encoded into visual characteristics (Belkin and Niyogi, 2003; Grinstein et al., 2002; Kromesch and Juhász, 2005; Singh et al., 2007), using statistical techniques such as Bayesian inference (Box and Tiao, 1992), multivariate classifier approaches like PCA (Abdi and Williams, 2010), self-organizing maps (Blanco et al., 2002; Ultsch, 2003), Random Forests (Breiman, 2001) and t-SNE (Maaten and Hinton, 2008), amongst others. Complicating issues of dimensionality and representation is the aspect of time-variance. Time can be mapped in static visualizations as another spatial dimension (as in simple 2D time-variant plots of 1D data), yet becomes more complex as dimensionality increases (Muller and Schumann, 2003). Time itself is complex: it can be discrete or continuous (time points vs time intervals), linear, cyclical, branching or multiperspective; it can be unidirectional or reversible, real or imaginary (Aigner et al., 2007). As with spatial or multivariate data, temporality demands an appropriate form of encoding, reduction and/or aggregation – in part driven by characteristics of the data and in part by the requirements of the visualization – including algorithmic as well as aesthetic/utilitarian considerations (Hao et al., 2005; Walker et al., 2016; Weber et al., 2001).

Graphics

At a fundamental level, data visualization transforms numerical data and accompanying metrics into visual representations, comprised of forms, colors, and their relationships (Wilkinson, 2005). Form can encode many features of data such as size, scale, shape, format, dimensionality, proximity, clustering and topology. Color-mapping (Crameri, 2018; Kovesi, 2015) can encode salient characteristics of data and entity-relationships as hue, saturation, and lightness, contingent upon their color-space

representation (Bujack et al., 2018). HD characteristics can be articulated through visual patterns, glyphs and other symbolic forms (Munzner, 2014).

However, the capabilities of computer graphics extend far beyond 2D/3D/HD graph-type primitives as representations of scalar, vector and tensor data (Hughes et al., 2013). Graphical entities can encode data representations in a huge range of visual variables such as textures and images, 3D models, volumetric entities and complex visual scenes (Hansen and Johnson, 2005; Telea, 2015). Consequently, computer graphics, whilst built up of programmatic primitives (e.g. pixels, lines, vertices, vector graphics, OpenGL entities, scenegraphs, shader graphs, lighting models), also iteratively participate in a chain of visualization operations of increasing abstraction throughout the workflow.

2.2 Interaction

The third stage in the workflow is situated broadly within Human-Computer Interaction (HCI) approaches (Dix, 2017). The disciplinary study of Human-Computer Interaction (HCI) operates at the intersection of computer science, behavioral and cognitive science, design and aesthetics (Campos et al., 2011; Myers, 1998). In the context of this research it concerns active work by the user of a visualization system, through interactive and animated interfaces.

Interactivity and animation

Interactivity enables the exploration of the design-space for visualization (Schulz et al., 2013), including constraints for visual encoding and interaction idioms (Munzner, 2014), creating a feedback loop between user and visualization system (Dimara and Perin, 2020; Glanville, 2007). Interactivity refers not only to the design and behavior of elements of a graphical-user interface (GUI) (Benyon, 2014; Shneiderman et al., 2016), but with the visualization itself, via activities such as cross-linked views and data selection and highlighting (Lam, 2008; Munzner, 2014; Nöllenburg, 2007). This includes the ability to zoom, pan and rotate displayed data, and adjust visual characteristics such as color, texture and lighting. GUIs and other interactive devices can be thought of as 'affordances' for users (Gibson, 1978; Stendal et al., 2016), where interface elements are suggestive of their utility as well as making certain actions possible (Chan et al., 2019; ElSayed et al., 2016). These affordances open up a large and complex parameter space for both visualization and interface design within a Model-View-Controller (MVC) architecture (Riehle, 1997; Seffah, 2015) that presents significant challenges for both design and technical implementation perspectives. Interactive interfaces are inherently animated or animatable, though consideration must be given to the utility and type of animation employed, with especial reference to human perception of motion and visual characteristics such as depth perception (Kruiger et al., 2017; Lowe, 2017).

Overview + detail

A key benefit of an interactive visualization versus a static image is articulated by Shneiderman's (1996) influential visual information-seeking 'mantra': overview first, zoom and filter, details on demand. This has been widely employed as a useful design guideline for user-interfaces, yet evaluation of its utility in practical implementations remains informal (Benyon, 2014; Cockburn et al., 2008). Visual Analytics adapts and modifies it to incorporate aspects of automation: analyze first, show the important; zoom, filter and analyze further – details on demand (Keim et al., 2006). This approach is pertinent for geospatial data, which can exhibit an extreme range of scales and consequently present challenges to simple synoptic visualizations. Interactive visualizations, in particular, facilitate examination in both overview and detail, forming an important part of the analytical and inferential processes (Keim, 2001; Keim et al., 2006) that actively facilitate conceptual model building and analyses (Harold et al., 2016; Keim et al., 2010; Ward et al., 2010).

2.3 Well-Posed Visualization

Efforts to model and evaluate 'effectiveness' of visualization remains both a problem of definition and measurement across a wide set of research domains (Behrisch et al., 2018; Keim et al., 2010; Zhu, 2007). 'Well-posed visualization' considers the relationship between human perception, the displayed data and the accuracy, utility and efficiency of visual representations as they are optimized for visual display (Zhu, 2007). This is particularly applicable to certain conventional approaches undertaken in geoscientific visualization, wherein, for example, extant techniques have a successful track record (e.g. time series, seismic cross-sections) but can be improved upon and augmented using new approaches.

Human perception

It is not productive to think of visualizations simply as pictures. The psychology and psychophysics of perception play a major role in visualization (Gibson, 1986; Gordon, 2004; Ware, 2013) and have a significant impact upon 'well-posedness'. Visual perception is a system in constant interaction within a context or environment. Gibson's (1986) *plenoptic function* proposes that we see the world not from a point but from a path, sampling the ambient array of light that falls upon the eye (Adelson and Bergen, 1991; McGinity, 2014). This is an important concept for data visualization, drawing attention to the fact that visualizations are always embedded in complex contexts, having implications for interaction and perception in the world and in immersive environments.

Perceptual and cognitive aspects of visualization are deeply interrelated (Bae et al., 2019), given that vision is a function of the human brain in response to its environment (Cavanagh, 2011; Stockman and Brainard, 2015; Thorpe et al., 1996). Visualization research

considers a range of human factors, such as visual search and selective attention (Müller and Krummenacher, 2006), the spatial resolution of visual attention (Intriligator and Cavanagh, 2001), shape perception, pre-attentive processing (Tory and Moller, 2004) and gist awareness (Oliva, 2005) amongst others.

Visual optimization

Visual optimization annexes data visualization and UI design to vision science, by optimizing interactive visual displays for human perception. Good visualizations account for and accommodate (maximize or minimize) salience, perceptual organization, visual clutter, gist extraction (visual summary), peripheral awareness, scene change and attentional blindness (Rosenholtz, 2017; Rosenholtz and Yu, 2019a, 2019b). Veridical perception is parameterized not only physiologically but also by task – in other words, what the purpose of a visualization is (Knill et al., 1996; Mark et al., 2010) and how it can be achieved efficiently (Chen and Golan, 2016).

Zhu (2007) summarizes effectiveness ('optimality') in three high-level principles: accuracy, utility and efficiency, as follows:

- Principle of Accuracy: For a visualization to be effective, the attributes of visual elements shall match the attributes of data items, and the structure of the visualization shall match the structure of the data set.
- Principle of Utility: An effective visualization should help users achieve the goal of specific tasks.
- Principle of Efficiency: An effective visualization should reduce the cognitive load for a specific task over non-visual representations.

'Well-posed' scientific visualization arises from the concurrent, managed realization of these principles in concert with a strong awareness of the role that human perception plays. It requires that a visualization maintains a clear objective/empirical relationship to the data that it represents, whilst taking into account human visual capabilities and limitations. This informs, for instance, the implementation of basic graphs and scientific figures, as they are direct graphical representations of data (Herman et al., 2000; Huang et al., 2009), yet this relationship becomes more ambiguous as data and data representations become more complex, abstract or increase in dimensionality (Munzner, 2014; Wang and Tao, 2017). This is not only an issue of design and implementation, but must take into account many aspects of human psychology, cognition and perception (Tversky, 2011; Ware, 2013).

Well-posed visualizations should also maintain an overt awareness of the risks of representational ambiguity and error and address concerns about reproducibility. This includes the capability to track data-flow provenance (Milton and Possolo, 2020; Silva et al., 2007) and to accurately redeploy or re-enact a visualization upon different substrates

(e.g. print) or hardware devices (e.g. screen, projector), with appropriately matched color values and visual geometry (Fairchild, 2013; Tufte, 1983; Ware, 2013). Reproducibility is a key consideration in terms of programmatic accuracy at a software level, and in software-hardware relationships. Given equivalently calibrated display systems, display environments, data and computational resources, scientists should be able to reproduce visualizations across systems and viewing/interaction situations. However, the workflows by which visualizations are made can become complex and difficult to reproduce (Belhajjame et al., 2012; Garijo et al., 2014). The ability to trace the constructive process of visualization can be assisted by software that saves logs of activity and applied values and parameters (e.g. how a data value is mapped to a color value, at what rate an animation is displayed, what sort of spatial mapping is deployed). This ability is specifically enhanced by dataflow diagrams and software that deploys procedural and dataflow programming methodologies – a model extensively deployed by scientific visualization, media production and game engine software (Chapter 3).

Reconnaissance and exploration

Fundamentals of exploratory computer-based data analysis are foreshadowed by Tukey (Tukey, 1990, 1977), with visualization approaches developing in parallel with technological capabilities over the following decades (Andrienko and Andrienko, 2006; Keim and Kriegel, 1996). Approaches from visual analytics (Keim et al., 2008; May et al., 2010) and data visualization (Battle and Heer, 2019) characterize exploration in a manner similar to information-seeking (Shneiderman, 1996). Crisan and Munzner (2019) provide the most pertinent characterization for this research: 'reconnaissance and task-wrangling are coordinated processes undertaken by domain experts to familiarize themselves with an unfamiliar data landscape'. A useful distinction is drawn between 'reconnaissance' as a high-level overview activity (exploring an unfamiliar data landscape) and 'investigative exploration' (a closer, more informed view) with tasks defined by a domain expert. Our workflow approaches this as an iterative activity, as indicated in Fig.2.

2.4 Knowledge

The fourth stage of the workflow consists of the knowledge generated throughout the workflow, characterized and conformed by its emergence through human-computer-data interaction and the evaluative criteria to which it is subjected. Epistemic actions (Ware, 2013) and feature observations created or enabled by the system are both empirical and semiotic in nature. The scientific case for semiotics is briefly introduced in Wilkinson (2005) and Zhu (2007), and discussed in detail in Ware (2013) and Tanaka-Ishii (2010). The evaluation of comparatively more useful visualizations – and associated inferential relevance - is contingent upon the analytical objectives and evaluative criteria for success

(Craft and Cairns, 2008; Johnson et al., 2006; Steele and Iliinsky, 2010).

Features

Visualizing features poses two initial questions of data: '*Is what we see really there?'* and '*Is there something there we cannot see?*'. The first question encapsulates the interplay between scientific curiosity and *apophenia* - 'the innate human ability to see pattern in noise' (Cook, 2017; Gauvrit et al., 2014; Wickham et al., 2010). The second question exposes the concept of 'missed discovery', where the analyst is unaware that unperceived structures await discovery (Buja et al., 2009; Mark et al., 2010). Visualization activities must take these countervailing forces into account, constructing veridical visualizations that assist identification of features in data (Cavanagh, 2011; Hinton, 2014; Kastens et al., 2016), and ally these discoveries with inference (Alhadad, 2018; Friston, 2003; Gooding, 2010). This process requires evaluation at a variety of levels of abstraction throughout the workflow, indicated by the associated loop in Fig.2.

Visual analytics and evaluation

Visual analytics (VA) aims to integrate the analytic capabilities of the computer and the abilities of the human analyst, through the development of software tools and visualization methodologies that explicitly incorporate the expert human user within the cooperative and recursive loops of machine-user analytical interaction (Keim et al., 2006; Sacha et al., 2014). The practical implementation of HCI within the context of interactive data visualization for scientific research (Dill et al., 2012; Keim et al., 2008) is an appealing model for the development of novel geoscientific tools, especially for those that deal with datasets that exhibit features for which human pattern-recognition excels (e.g. object recognition, high-order grouping, gist representation (Oliva, 2005)) and in which human expert observation can be articulated within the data-inference process (Borkin et al., 2016; Pohlmeyer et al., 2011). Concepts and techniques from VA that build upon explicit domain knowledges suggest future developmental pathways for geoscientific visualization research (Federico et al., 2017).

2.5 Summary

This chapter has provided an overview of the broad conceptual background underpinning this research. Geoscientists and other data scientists routinely use software packages to visualize and present results as static graphs and charts. This graphing and underpinning 'spreadsheet' document model is user-friendly, robust and has been in use for hundreds of years (Campbell-Kelly, 2003), with more recent computer-based implementations that enable dynamic updating of variables and display capabilities. Whilst adequate for many scientific purposes, this approach is obviously limited in scope

and can lead to missed insights, as datasets increase in size and complexity. As an alternative, many users engaged in scientific research are comfortable creating visual output using programming models and scripting (e.g. Python, R, Matlab) as datasets become larger. One benefit of this is that visual outputs can illuminate some of the dynamic aspects of underpinning data, such as via animation for temporal data. Similarly, complex three-dimensional surfaces can be interacted-with and observed from different perspectives, rather than being seen only from a fixed point of view in a static picture. However, decisions must be made about the effectiveness and utility of interactivity and animation for scientific purposes. What constitutes 'good' and 'bad' visualization and when interactivity is useful or unnecessary, or even obfuscating, is the subject of ongoing research (Few, 2015b, 2015a; Tufte, 1997; Victor, 2005). What constitutes effective or optimal interactive visualization for geoscientific inference is explored in the research described herein.

Interactive, animated representations of data enable sophisticated possibilities for interrogation. With the advent of high performance computer graphics, the repertoire of graph types and visualizations has significantly expanded (Beck et al., 2017; Herman et al., 2000; Liu et al., 2017, 2014) including developments for novel display environments such as XR, ' Extended Reality', adumbrating VR ('Virtual Reality'), AR ('Augmented Reality'), MR ('Mixed Reality') (Billinghurst et al., 2015; Costanza et al., 2009) and Dome display systems (Bourke, 2009; Fonnet and Prié, 2019; Kwasnitschka, 2017). These displays and associated programming toolsets, such as procedural graphics and computer game engines, have been relatively under-explored as platforms for geoscientific data visualization. They provide affordances that engage the human sensorium in novel ways, and suggest exciting possibilities for analytical and inferential visualization in the Earth and planetary sciences.

2.6 References

Abdi, H., Williams, L.J., 2010. Principal Component Analysis, in: Principal Component Analysis, Wiley Interdisciplinary Reviews: Computational Statistics. Wiley Online Library.

Adelson, E.H., Bergen, J.R., 1991. The plenoptic function and the elements of early vision. Computational models of visual processing 1.

Aigner, W., Miksch, S., Müller, W., Schumann, H., Tominski, C., 2007. Visualizing time-oriented data—A systematic view. Computers & Graphics 31, 401–409. https://doi.org/10.1016/j.cag.2007.01.030

Alhadad, S., 2018. Visualizing Data to Support Judgement, Inference, and Decision Making in Learning Analytics: Insights from Cognitive Psychology and Visualization Science. Journal of Learning Analytics 5, 60–85. https://doi.org/10/gf2qrt

Andrienko, N., Andrienko, G., 2006. Exploratory Analysis of Spatial and Temporal Data: A Systematic Approach. Springer Science & Business Media.

Bae, J., Falkman, G., Helldin, T., Riveiro, M., 2019. Visual Data Analysis, in: Said, A., Torra, V. (Eds.), Data Science in Practice. Springer International Publishing, Cham, pp. 133–155. https://doi.org/10.1007/978-3-319-97556-6_8

Battle, L., Heer, J., 2019. Characterizing Exploratory Visual Analysis: A Literature Review and Evaluation of Analytic Provenance in Tableau. Computer Graphics Forum 38, 145–159. https://doi.org/10/ggjbvh

Beck, F., Burch, M., Diehl, S., Weiskopf, D., 2017. A Taxonomy and Survey of Dynamic Graph Visualization. Computer Graphics Forum 36, 133–159. https://doi.org/10/f9wkd9

Behrisch, M., Blumenschein, M., Kim, N.W., Shao, L., El-Assady, M., Fuchs, J., Seebacher, D., Diehl, A., Brandes, U., Pfister, H., Schreck, T., Weiskopf, D., Keim, D.A., 2018. Quality Metrics for Information Visualization. Computer Graphics Forum 37, 625–662. https://doi.org/10.1111/cgf.13446

Belhajjame, K., Corcho, O., Garijo, D., Zhao, J., Missier, P., Newman, D.R., Palma, R., Bechhofer, S., Garcia-Cuesta, E., Gomez-Perez, J.M., 2012. Workflow-Centric Research Objects: A First Class Citizen in the Scholarly Discourse., in: SePublica@ ESWC. pp. 1–12.

Belkin, M., Niyogi, P., 2003. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. Neural Computation 15, 1373–1396. https://doi.org/10.1162/089976603321780317

Benyon, D., 2014. Designing Interactive Systems: A comprehensive guide to HCI, UX and interaction design, 3rd Edition. ed. Pearson Education.

Billinghurst, M., Clark, A., Lee, G., 2015. A Survey of Augmented Reality. FNT in Human–Computer Interaction 8, 73–272. https://doi.org/10/gc2hkv

Blanco, I.D., Vega, A.A.C., González, A.B.D., 2002. Correlation Visualization of High Dimensional Data Using Topographic Maps, in: Dorronsoro, J.R. (Ed.), Artificial Neural Networks — ICANN 2002. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 1005–1010. https://doi.org/10.1007/3-540-46084-5_163

Borkin, M.A., Bylinskii, Z., Kim, N.W., Bainbridge, C.M., Yeh, C.S., Borkin, D., Pfister, H., Oliva, A., 2016. Beyond Memorability: Visualization Recognition and Recall. IEEE Transactions on Visualization and Computer Graphics 22, 519–528. https://doi.org/10.1109/TVCG.2015.2467732

Bourke, P., 2009. Omni-directional Stereoscopic Fisheye Images for Immersive Hemispherical Dome Environments, in: 2nd Annual International Conferences on Computer Games, Multimedia and Allied Technology (CGAT 2009). Presented at the Annual International Conferences on Computer Games, Multimedia and Allied Technology, Global Science & Technology Forum (GSTF). https://doi.org/10/ggswzp

Box, G.E.P., Tiao, G.C., 1992. Bayesian Inference in Statistical Analysis, Wiley Classics Library Edition. John Wiley & Sons.

Breiman, L., 2001. Random forests. Machine learning 45, 5–32. https://doi.org/10.1023/A:1010933404324

Buja, A., Cook, D., Hofmann, H., Lawrence, M., Lee, E.-K., Swayne, D.F., Wickham, H., 2009. Statistical inference for exploratory data analysis and model diagnostics. Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences 367, 4361–4383. https://doi.org/10.1098/rsta.2009.0120

Bujack, R., Turton, T.L., Samsel, F., Ware, C., Rogers, D.H., Ahrens, J., 2018. The Good, the Bad, and the Ugly: A Theoretical Framework for the Assessment of Continuous Colormaps. IEEE Transactions on Visualization and Computer Graphics 24, 923–933. https://doi.org/10.1109/TVCG.2017.2743978

Campbell-Kelly, M., 2003. The history of mathematical tables: from Sumer to spreadsheets. Oxford University Press.

Campos, P., Graham, N., Jorge, J., Nunes, N., Palanque, P., Winckler, M., 2011. Human-computer Interaction–INTERACT 2011: 13th IFIP TC 13 International Conference, Lisbon, Portugal, September 5-9, 2011, Proceedings. Springer.

Carlsson, G., 2009. Topology and data. Bulletin of the American Mathematical Society 46, 255–308. https://doi.org/10/c474bm

Cavanagh, P., 2011. Visual cognition. Vision Research 51, 1538–1551. https://doi.org/10.1016/j.visres.2011.01.015

Chan, J.K.Y., Wong, R.Y.M., Cheung, C.M.K., 2019. How Affordances of Immersive Visualization Systems Affect Learning Outcomes through Aesthetic Experience 9.

Chang, C., Dwyer, T., Marriott, K., 2018. An Evaluation of Perceptually Complementary Views for Multivariate Data, in 2018 IEEE Pacific Visualization Symposium (PacificVis). Presented at the 2018 IEEE Pacific Visualization Symposium (PacificVis), pp. 195–204. https://doi.org/10/gfw9mx

Chen, M., Golan, A., 2016. What May Visualization Processes Optimize? IEEE Trans. Visual. Comput. Graphics 22, 2619–2632. https://doi.org/10/f8959w

Chen, M., Hauser, H., Rheingans, P., Scheuermann, G. (Eds.), 2020. Foundations of Data Visualization. Springer International Publishing, Cham. https://doi.org/10.1007/978-3-030-34444-3

Chen, M.-S., Han, J., Yu, P.S., 1996. Data mining: an overview from a database perspective. Knowledge and data Engineering, IEEE Transactions on 8, 866–883. https://doi.org/10.1109/69.553155

Cockburn, A., Karlson, A., Bederson, B.B., 2008. A review of overview+detail, zooming, and focus+context interfaces. ACM Computing Surveys 41, 1–31. https://doi.org/10.1145/1456650.1456652

Cook, D., 2017. Myth busting and apophenia in data visualisation: is what you see really there? | [WWW Document]. URL http://dicook.org/talk/ihaka/ (accessed 2.12.19).

Costanza, E., Kunz, A., Fjeld, M., 2009. Mixed Reality: A Survey, in: Lalanne, D., Kohlas, J. (Eds.), Human Machine Interaction. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 47–68. https://doi.org/10.1007/978-3-642-00437-7_3

Craft, B., Cairns, P., 2008. Directions for Methodological Research in Information Visualization, in 2008 12th International Conference Information Visualisation. Presented at the 2008 12th International Conference Information Visualisation (IV), IEEE, London, UK, pp. 44–50. https://doi.org/10.1109/IV.2008.88

Crameri, F., 2018. Scientific colour maps. Zenodo, Oslo, Norway. https://doi.org/10.5281/zenodo.1243862

Crisan, A., Munzner, T., 2019. Uncovering Data Landscapes through Data Reconnaissance and Task Wrangling, in: 2019 IEEE Visualization Conference (VIS). Presented at the 2019 IEEE Visualization Conference (VIS), IEEE, Vancouver, BC, Canada, pp. 46–50. https://doi.org/10/gg64cv

Cunningham, A., Xu, K., Thomas, B., 2010. Seeing more than the graph: evaluation of multivariate graph visualization methods, in: Proceedings of the International Conference on Advanced Visual Interfaces - AVI '10. Presented at the the International Conference, ACM Press, Roma, Italy, p. 429. https://doi.org/10.1145/1842993.1843106

Dill, J., Earnshaw, R., Kasik, D., Vince, J., Wong, P.C. (Eds.), 2012. Expanding the Frontiers of Visual Analytics and Visualization. Springer London, London. https://doi.org/10.1007/978-1-4471-2804-5

Dimara, E., Perin, C., 2020. What is Interaction for Data Visualization? IEEE Trans. Visual. Comput. Graphics 26, 119–129. https://doi.org/10/ggbwhz

Dix, A., 2017. Human–computer interaction, foundations and new paradigms. Journal of Visual Languages & Computing 42, 122–134. https://doi.org/10.1016/j.jvlc.2016.04.001

ElSayed, N.A.M., Thomas, B.H., Marriott, K., Piantadosi, J., Smith, R.T., 2016. Situated Analytics: Demonstrating immersive analytical tools with Augmented Reality. Journal of Visual Languages & Computing 36, 13–23. https://doi.org/10.1016/j.jvlc.2016.07.006

Fairchild, M.D., 2013. Color Appearance Models Third Edition. John Wiley & Sons.

Fayyad, U., 2001. The digital physics of data mining. Communications of the ACM 44, 62–65. https://doi.org/10.1145/365181.365198

Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., 1996a. The KDD process for extracting useful knowledge from volumes of data. Communications of the ACM 39, 27–34. https://doi.org/10.1145/240455.240464

Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., 1996b. From data mining to knowledge discovery in databases. AI magazine 17, 37.

Fayyad, U.M., Smyth, P., 1999. Cataloging and Mining Massive Datasets for Science Data Analysis. Journal of Computational and Graphical Statistics 8, 589. https://doi.org/10.2307/1390878

Federico, P., Wagner, M., Rind, A., Amor-Amorós, A., Miksch, S., Aigner, W., 2017. The Role of Explicit Knowledge: A Conceptual Model of Knowledge-Assisted Visual Analytics, in: Proc. IEEE Conf. Visual Analytics Science and Technology. Presented at the IEEE VAST 2017, p. 12. https://doi.org/10.1109/VAST.2017.8585498

Few, S., 2015a. Signal: Understanding what Matters in a World of Noise. Analytics Press.

Few, S., 2015b. Information Visualization Research as Pseudo-Science [WWW Document]. URL http://perceptualedge.com/articles/visual_business_intelligence/infovis_research_as_pseudo-science.pdf (accessed 9.2.20).

Few, S., 2009. Now you see it: simple visualization techniques for quantitative analysis. Analytics Press.

Fonnet, A., Prié, Y., 2019. Survey of Immersive Analytics. IEEE Transactions on Visualization and Computer Graphics 1–1. https://doi.org/10/gf73gm

Fox, P., Hendler, J., 2011. Changing the Equation on Scientific Data Visualization. Science 331, 705–708. https://doi.org/10.1126/science.1197654

Friston, K., 2003. Learning and inference in the brain. Neural Networks 16, 1325–1352. https://doi.org/10/cqzvxd

Garijo, D., Alper, P., Belhajjame, K., Corcho, O., Gil, Y., Goble, C., 2014. Common motifs in scientific workflows: An empirical analysis. Future Generation Computer Systems 36, 338–351. https://doi.org/10/f55h9v

Gauvrit, N., Soler-Toscano, F., Zenil, H., 2014. Natural scene statistics mediate the perception of image complexity. Visual Cognition 22, 1084–1091. https://doi.org/10.1080/13506285.2014.950365

Gibson, J.J., 1986. The ecological approach to visual perception. Psychology Press, New York, NY.

Gibson, J.J., 1978. The Ecological Approach to the Visual Perception of Pictures. Leonardo 11, 227. https://doi.org/10.2307/1574154

Gil, Y., Deelman, E., Ellisman, M., Fahringer, T., Fox, G.C., Gannon, D., Goble, C., Livny, M., Moreau, L., Myers, J., 2007. Examining the Challenges of Scientific Workflows. Computer 40, 24–32. https://doi.org/10.1109/MC.2007.421

Glanville, R., 2007. Try again. Fail again. Fail better: The cybernetics in design and the design in cybernetics. Kybernetes. https://doi.org/10/dtqvdq

Gooding, D.C., 2010. Visualizing Scientific Inference. Topics in Cognitive Science 2, 15–35. https://doi.org/10/fpjfwk

Gordon, I.E., 2004. Theories of visual perception, 3. ed. ed. Psychology Press, Hove.

Gray, J., Liu, D.T., Nieto-Santisteban, M., Szalay, A., DeWitt, D.J., Heber, G., 2005. Scientific data management in the coming decade. ACM SIGMOD Record 34, 34–41. https://doi.org/10.1145/1107499.1107503

Grinstein, G.G., Hoffman, P., Pickett, R.M., Laskowski, S.J., 2002. Benchmark development for the evaluation of visualization for data mining. Information visualization in data mining and knowledge discovery 129–176.

Hansen, C.D., Johnson, C.R. (Eds.), 2005. The visualization handbook. Elsevier-Butterworth Heinemann, Amsterdam ; Boston.

Hao, M.C., Dayal, U., Keim, D.A., Schreck, T., 2005. Importance-driven visualization layouts for large time series data, in: Information Visualization, 2005. INFOVIS 2005. IEEE Symposium On. IEEE, pp. 203–210. https://doi.org/10/cttw3x

Harold, J., Lorenzoni, I., Shipley, T.F., Coventry, K.R., 2016. Cognitive and psychological science insights to improve climate change data visualization. Nature Climate Change 6, 1080–1089. https://doi.org/10.1038/nclimate3162

He, X., Tao, Y., Wang, Q., Lin, H., 2019. Multivariate Spatial Data Visualization: A Survey. J Vis 22, 897–912. https://doi.org/10/ggnsfw

Herman, I., Melancon, G., Marshall, M.S., 2000. Graph visualization and navigation in information visualization: A survey. IEEE Transactions on Visualization and Computer Graphics 6, 24–43. https://doi.org/10.1109/2945.841119

Hey, T., Tansley, S., Tolle, K.M., others, 2009. The fourth paradigm: data-intensive scientific discovery. Microsoft research Redmond, WA.

Hinton, G., 2014. Where Do Features Come From? Cognitive Science 38, 1078–1101. https://doi.org/10.1111/cogs.12049

Huang, W., Eades, P., Hong, S.-H., 2009. Measuring Effectiveness of Graph Visualizations: A Cognitive Load Perspective. Information Visualization 8, 139–152. https://doi.org/10.1057/ivs.2009.10

Hughes, J.F., Van Dam, A., Foley, J.D., McGuire, M., Feiner, S.K., Sklar, D.F., Akeley, K., 2013. Computer graphics: principles and practice. Addison-Wesley Professional.

Inselberg, A., 2009. Parallel Coordinates: Visual Multidimensional Geometry and its Applications. Springer.

Inselberg, A., 2006. The Promise and Challenge of Multidimensional Visualization, in: Etzion, O., Kuflik, T., Motro, A. (Eds.), Next Generation Information Technologies and Systems. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 355–356. https://doi.org/10.1007/11780991_36

Intriligator, J., Cavanagh, P., 2001. The Spatial Resolution of Visual Attention. Cognitive Psychology 43, 171–216. https://doi.org/10.1006/cogp.2001.0755

Johnson, C., Moorhead, R., Munzner, T., Pfister, H., Rheingans, P., Yoo, T.S., 2006. NIH/NSF visualization research challenges report, in: Los Alamitos, Ca: IEEE Computing Society. Citeseer. https://doi.org/10.1109/MCSE.2006.77

Kastens, K.A., Shipley, T.F., Boone, A.P., Straccia, F., 2016. What Geoscience Experts and Novices Look At, and What They See, When Viewing Data Visualizations. Journal of Astronomy & Earth Sciences Education 3, 27–58. https://doi.org/10/gf4ksr

Kehrer, J., Hauser, H., 2013. Visualization and Visual Analysis of Multifaceted Scientific Data: A Survey. IEEE Transactions on Visualization and Computer Graphics 19, 495–513. https://doi.org/10.1109/TVCG.2012.110

Keim, D., 2001. Visual exploration of large data sets. Communications of the ACM 44, 38–44. https://doi.org/10.1145/381641.381656

Keim, D., Andrienko, G., Fekete, J.-D., Görg, C., Kohlhammer, J., Melançon, G., 2008. Visual Analytics: Definition, Process, and Challenges, in: Kerren, A., Stasko, J.T., Fekete, J.-D.,

North, C. (Eds.), Information Visualization. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 154–175. https://doi.org/10.1007/978-3-540-70956-5_7

Keim, D., Kohlhammer, J., Ellis, G., Mansmann, F., 2010. Mastering the information age solving problems with visual analytics. Eurographics Association.

Keim, D., Kriegel, H.-P., 1996. Visualization techniques for mining large databases: a comparison. IEEE Transactions on Knowledge and Data Engineering 8, 923–938. https://doi.org/10.1109/69.553159

Keim, D., Mansmann, F., Schneidewind, J., Ziegler, H., 2006. Challenges in Visual Data Analysis, in: Tenth International Conference on Information Visualisation (IV'06). Presented at the Tenth International Conference on Information Visualisation (IV'06), IEEE, London, England, pp. 9–16. https://doi.org/10.1109/IV.2006.31

Knill, D.C., Kersten, D., Yuille, A., 1996. Introduction: A Bayesian formulation of visual perception. Perception as Bayesian inference 1–21. https://doi.org/10.1017/CBO9780511984037.002

Kovesi, P., 2015. Good Colour Maps: How to Design Them. arXiv:1509.03700 [cs]. https://doi.org/10.1071/ASEG2015ab107

Kromesch, S., Juhász, S., 2005. High dimensional data visualization, in: 6th International Symposium of Hungarian Researchers on Computational Intelligence. Citeseer, pp. 1–12.

Kruiger, J., Hassoumi, A., Schulz, H.-J., Telea, A., Hurter, C., 2017. Multidimensional Data Exploration by Explicitly Controlled Animation. Informatics 4, 26. https://doi.org/10.3390/informatics4030026

Kwasnitschka, T., 2017. Planetariums — not just for kids. Nature 544, 395–395. https://doi.org/10/gf4ksh

Lam, H.L.M., 2008. Visual Exploratory Analysis of Large Data Sets Evaluation and Application (PhD Thesis). Citeseer.

Lau, A., Vande Moere, A., 2007. Towards a Model of Information Aesthetics in Information Visualization, in: 2007 11th International Conference Information Visualization (IV '07). Presented at the 2007 11th International Conference Information Visualization (IV '07), IEEE, Zurich, Switzerland, pp. 87–92. https://doi.org/10.1109/IV.2007.114

Liu, S., Cui, W., Wu, Y., Liu, M., 2014. A survey on information visualization: recent advances and challenges. The Visual Computer 30, 1373–1393. https://doi.org/10.1007/s00371-013-0892-3

Liu, S., Maljovec, D., Wang, B., Bremer, P.-T., Pascucci, V., 2017. Visualizing High-Dimensional Data: Advances in the Past Decade. IEEE Transactions on Visualization and Computer Graphics 23, 1249–1268. https://doi.org/10.1109/TVCG.2016.2640960

Lowe, R., 2017. Learning from dynamic visualization. Springer Berlin Heidelberg, New York, NY.

Maaten, L.V.D., Hinton, G.E., 2008. Visualizing Data using t-SNE. Journal of Machine Learning Research 9, 2579–2605. https://doi.org/10.1007/s10479-011-0841-3

MacEachren, A.M., 2001. An evolving cognitive-semiotic approach to geographic visualization and knowledge construction. Information Design Journal 10, 26–36. https://doi.org/10.1075/idj.10.1.06mac

Mark, J.T., Marion, B.B., Hoffman, D.D., 2010. Natural selection and veridical perceptions. Journal of Theoretical Biology 266, 504–515. https://doi.org/10.1016/j.jtbi.2010.07.020

May, R., Hanrahan, P., Keim, D.A., Shneiderman, B., Card, S., 2010. The state of visual analytics: views on what visual analytics is and where it is going, in: Visual Analytics Science and Technology (VAST), 2010 IEEE Symposium On. IEEE, pp. 257–259. https://doi.org/10.1109/VAST.2010.5649078

McGinity, M., 2014. Presence, Immersion and the Panorama (PhD Thesis). University of New South Wales, Sydney, Australia.

Milton, M.J.T., Possolo, A., 2020. Trustworthy data underpin reproducible research. Nature Physics 16, 117–119. https://doi.org/10/ggktsr

Müller, H.J., Krummenacher, J., 2006. Visual search and selective attention. Visual Cognition 14, 389–410. https://doi.org/10.1080/13506280500527676

Muller, W., Schumann, H., 2003. Visualization methods for time-dependent data - an overview, in: Proceedings of the 2003 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.03EX693). Presented at the 2003 Winter Simulation Conference, IEEE, New Orleans, LA, USA, pp. 737–745. https://doi.org/10.1109/WSC.2003.1261490

Munzner, T., 2014. Visualization Analysis and Design. A K Peters/CRC Press. https://doi.org/10.1201/b17511

Myers, B.A., 1998. A brief history of human-computer interaction technology. interactions 5, 44–54. https://doi.org/10.1145/274430.274436

Nöllenburg, M., 2007. Geographic visualization, in: Kerren, A. (Ed.), Human-Centered Visualization Environments, LLNC 4417. Springer, Berlin Heidelberg, pp. 257–294.

Oliva, A., 2005. Gist of the scene. Neurobiology of attention 696, 251–258. https://doi.org/10.1016/B978-012375731-9/50045-8

Pearl, J., 2009. Causality: Models, Reasoning, And Inference. Cambridge University Press (Cambridge).

Pohlmeyer, E.A., Wang, J., Jangraw, D.C., Lou, B., Chang, S.-F., Sajda, P., 2011. Closing the loop in cortically-coupled computer vision: a brain–computer interface for searching image databases. Journal of Neural Engineering 8, 036025. https://doi.org/10.1088/1741-2560/8/3/036025

Popper, K., 1959. The Logic of Scientific Discovery. Routledge.

Purchase, H.C., Andrienko, N., Jankun-Kelly, T.J., Ward, M., 2008. Theoretical Foundations of Information Visualization, in: Kerren, A., Stasko, J.T., Fekete, J.-D., North, C. (Eds.), Information Visualization. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 46–64. https://doi.org/10.1007/978-3-540-70956-5_3

Reina, G., Childs, H., Matković, K., Bühler, K., Waldner, M., Pugmire, D., Kozlíková, B., Ropinski, T., Ljung, P., Itoh, T., Gröller, E., Krone, M., 2020. The moving target of visualization software for an increasingly complex world. Computers & Graphics 87, 12–29. https://doi.org/10/ggk8w4

Riehle, D., 1997. Composite design patterns. ACM SIGPLAN Notices 32, 218–228. https://doi.org/10/bq66j2

Rosenholtz, R., 2017. Capacity limits and how the visual system copes with them. Electronic Imaging 2017, 8–23. https://doi.org/10/gg63hz

Rosenholtz, R., Yu, D., 2019a. Modern Vision Science for Designers: Making Designs Clear at a Glance, in: Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems. Presented at the CHI '19: CHI Conference on Human Factors in Computing Systems, ACM, Glasgow Scotland UK, pp. 1–5. https://doi.org/10/gg63hw

Rosenholtz, R., Yu, D., 2019b. Human Vision at a Glance, in: 14th International Conference on Spatial Information Theory. Presented at the COSIT 2019, Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany, Schloss Dagstuhl, p. 4. https://doi.org/10/gg7tj6

Sacha, D., Stoffel, A., Stoffel, F., Kwon, B.C., Ellis, G., Keim, D.A., 2014. Knowledge Generation Model for Visual Analytics. IEEE Transactions on Visualization and Computer Graphics 20, 1604–1613. https://doi.org/10/f6qj6x

Schulz, H., Nocke, T., Heitzler, M., Schumann, H., 2013. A Design Space of Visualization Tasks. IEEE Transactions on Visualization and Computer Graphics 19, 2366–2375. https://doi.org/10.1109/TVCG.2013.120

Seffah, A., 2015. Patterns of hci design and hci design of patterns: bridging HCI design and model-driven software engineering. Springer Berlin Heidelberg, New York, NY.

Sellars, S., Nguyen, P., Chu, W., Gao, X., Hsu, K., Sorooshian, S., 2013. Computational Earth Science: Big Data Transformed into Insight. Eos, Transactions American Geophysical Union 94, 277–278. https://doi.org/10.1002/2013EO320001

Shneiderman, B., 1996. The eyes have it: a task by data type taxonomy for information visualizations, in: Proceedings 1996 IEEE Symposium on Visual Languages. Presented at the Proceedings 1996 IEEE Symposium on Visual Languages, pp. 336–343. https://doi.org/10/fwdq26

Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., Diakopoulos, N., 2016. Designing the User Interface: Strategies for Effective Human-Computer Interaction, 6th ed. Pearson.

Silva, C.T., Freire, J., Callahan, S.P., 2007. Provenance for Visualizations: Reproducibility and Beyond. Computing in Science & Engineering 9, 82–89. https://doi.org/10.1109/MCSE.2007.106

Singh, G., Mémoli, F., Carlsson, G.E., 2007. Topological Methods for the Analysis of High Dimensional Data Sets and 3D Object Recognition., in: SPBG. pp. 91–100.

Snyder, J., 1987. Map Projections- A Working Manual, U.S. Geological Survey professional paper; 1395). U.S. Geological Survey.

Steele, J., Iliinsky, N.P.N. (Eds.), 2010. Beautiful visualization: looking at data through the eyes of experts, 1st ed. ed. O'Reilly, Sebastopol, CA.

Stendal, K., Thapa, D., Lanamaki, A., 2016. Analyzing the Concept of Affordances in Information Systems, in: 2016 49th Hawaii International Conference on System Sciences (HICSS). Presented at the 2016 49th Hawaii International Conference on System Sciences (HICSS), IEEE, Koloa, HI, USA, pp. 5270–5277. https://doi.org/10/gg64b9

Stockman, A., Brainard, D.H., 2015. Fundamentals of color vision I: color processing in the eye, in: Elliot, A.J., Fairchild, M.D., Franklin, A. (Eds.), Handbook of Color Psychology. Cambridge University Press, Cambridge, pp. 27–69. https://doi.org/10.1017/CBO9781107337930.004

Tanaka-Ishii, K., 2010. Semiotics of Programming. Cambridge University Press.

Telea, A.C., 2015. Data visualization: principles and practice. CRC Press.

Thorpe, S., Fize, D., Marlot, C., 1996. Speed of processing in the human visual system. Nature 381, 520–2. https://doi.org/10.1038/381520a0

Tory, M., Moller, T., 2004. Human factors in visualization research. IEEE Transactions on Visualization and Computer Graphics 10, 72–84. https://doi.org/10.1109/TVCG.2004.1260759

Tufte, E.R., 1997. Visual explanations: images and quantities, evidence and narrative. Graphics Press, Cheshire, CT.

Tufte, E.R., 1990. Envisioning information. Graphics Press, Cheshire, CT, USA.

Tufte, E.R., 1983. The visual display of quantitative information. Graphics press Cheshire, CT.

Tukey, J.W., 1990. Data-Based Graphics: Visual Display in the Decades to Come. Statistical Science 5, 327–339.

Tukey, J.W., 1977. Exploratory data analysis. Reading, Mass.

Tversky, B., 2011. Visualizing Thought: Topics in Cognitive Science. Topics in Cognitive Science 3, 499–535. https://doi.org/10.1111/j.1756-8765.2010.01113.x

Ultsch, A., 2003. Maps for the visualization of high-dimensional data spaces, in: Proc. Workshop on Self Organizing Maps. pp. 225–230.

Van Wijk, J.J., 2005. The value of visualization, in: Visualization, 2005. VIS 05. IEEE. IEEE, pp. 79–86.

Victor, B., 2005. Magic Ink: Information Software and the Graphical Interface [WWW Document]. URL http://worrydream.com/MagicInk/ (accessed 2.14.20).

Walker, J., Borgo, R., Jones, M.W., 2016. TimeNotes: A Study on Effective Chart Visualization and Interaction Techniques for Time-Series Data. IEEE Transactions on Visualization and Computer Graphics 22, 549–558. https://doi.org/10.1109/TVCG.2015.2467751

Wang, C., Tao, J., 2017. Graphs in scientific visualization: A survey, in: Computer Graphics Forum. Wiley Online Library, pp. 263–287. https://doi.org/10.1111/cgf.12800

Ward, M.O., Grinstein, G., Keim, D., 2010. Interactive data visualization: foundations, techniques, and applications. AK Peters/CRC Press, Natick, Massachusetts.

Ware, C., 2013. Information visualization: perception for design, 3rd ed. Elsevier, Amsterdam, Holland.

Weber, M., Alexa, M., Muller, W., 2001. Visualizing time-series on spirals, in: IEEE Symposium on Information Visualization, 2001. INFOVIS 2001. Presented at the IEEE Symposium on Information Visualization, 2001. INFOVIS 2001., IEEE, San Diego, Ca, USA, pp. 7–13. https://doi.org/10.1109/INFVIS.2001.963273

Wickham, H., Cook, D., Hofmann, H., Buja, A., 2010. Graphical inference for infovis. IEEE Transactions on Visualization and Computer Graphics 16, 973–979. https://doi.org/10.1109/TVCG.2010.161

Wilkinson, L., 2005. The grammar of graphics, 2nd ed. Springer Science & Business Media, Chicago, IL, USA.

Wong, P.C., Bergeron, R.D., 1994. 30 years of multidimensional multivariate visualization. Scientific Visualization 2, 3–33.

Zhu, Y., 2007. Measuring effective data visualization, in: Advances in Visual Computing. Springer, pp. 652–661.

Chapter 3: Technical Review

3 Introduction

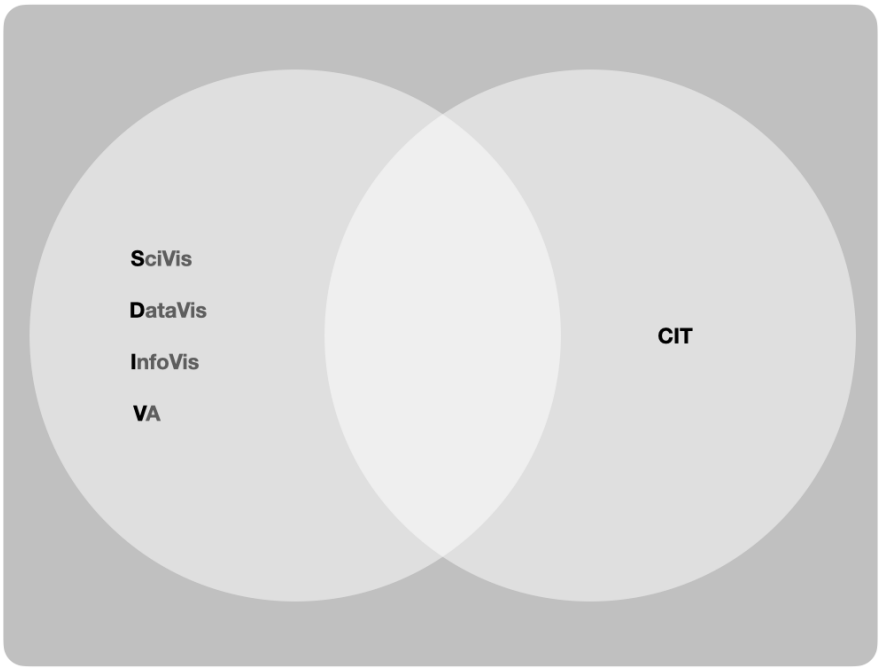


Fig. 1: Visualization disciplines SDIV and CIT, as outlined in text

This chapter reviews the technical and design components that are candidates for inclusion in a system that implements the visualization workflow, incorporating interactivity and an understanding of human perception, as outlined in Chapter 2.

The disciplines of Scientific Visualization (SciVis), Data Visualization (DataVis), Information Visualization (InfoVis) and Visual Analytics (VA) expose applicable technical approaches and design insights (Bae et al., 2019; Brodbeck et al., 2009; Chen et al., 2020; Purchase et al., 2008). They share many perceptual imperatives, design features, epistemic actions, technical aspects, and algorithms (Munzner, 2014; Telea, 2015; Ward et al., 2010; Ware, 2013). They are grouped in this technical outline as SDIV.

As a key extension to hardware and software environments that are likely to be familiar to scientific users, this research explores technical and design approaches enabled by interactive media, computer game engines and digital cinema, irrespective of disciplinary distinctions (Fig.1). Hereafter grouped as 'Creative Industry Technologies' (Abbasi et al., 2017; Hartley, 2005), CIT consolidates a variety of hardware and software methods that do not normally fall within the purview of individual disciplinary methodologies or toolsets. For instance, digital display systems that utilize CIT, developed within the galleries, libraries, archives and museums (GLAM) sector (Kenderdine, 2010; Lewi et al., 2019), have made notable connections across disciplinary communities and exposed new modalities for data reconnaissance and exploration (Bourke, 2018, 2017). Great benefit is to be found in exploring the technical and design commonalities of these approaches.

Interactive data visualization pipeline

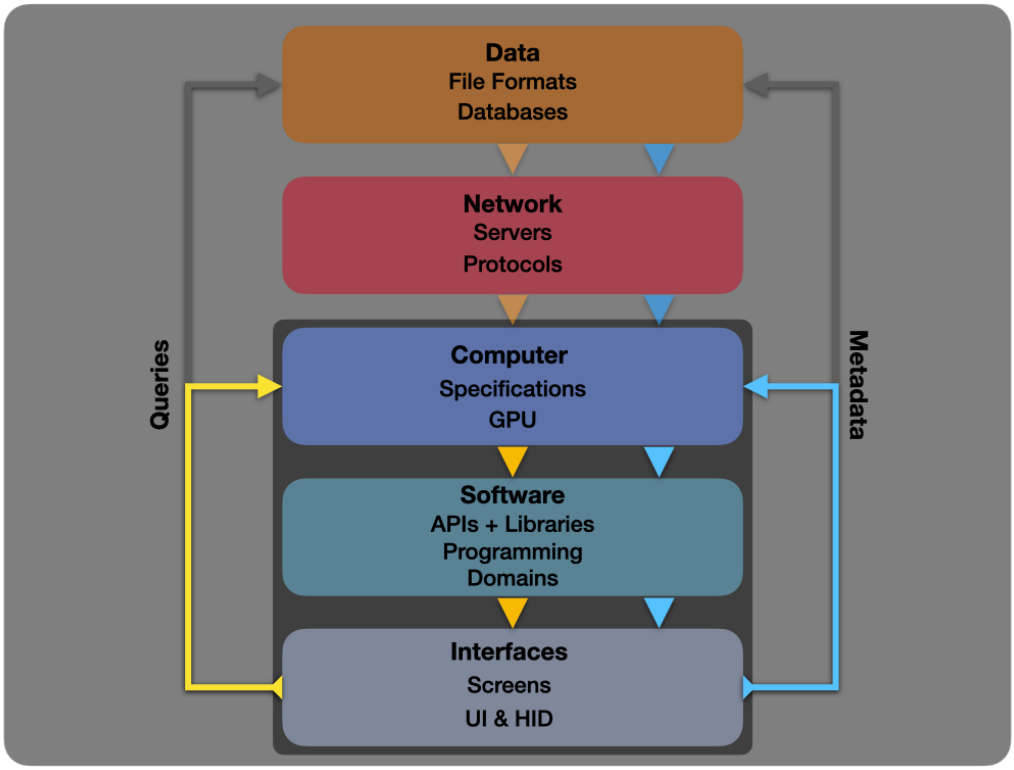


Fig. 2: Interactive Visualization Pipeline

The structure of an interactive data visualization system can be thought of as a pipeline with back-propagation. Fig. 2 provides a high-level abstraction, breaking the system into five main components: Data, Network, Computer, Software and Interfaces. These are connected by back-propagation feedback for queries and metadata flow. The rest of this chapter is structured according to this model, briefly introducing the first two components, with a main focus upon the final three components of the pipeline:

3.1 Data. The first component concerns basic forms of data storage and their associated filetypes. This includes various types of databases and their capabilities.

3.2 Network. The second component concerns read/write access to data repositories over a network and the protocols used.

3.3 Computer. The third component details general hardware specifications and Graphical Processing Unit (GPU) capabilities.

3.4 Software. The fourth component concerns graphics APIs and libraries, and representative software choices for interactive visualization grouped according to the SDIV/CIT schema (Fig.1). It introduces dataflow and visual programming approaches.

3.5 Interfaces. The fifth component considers types of screen displays, including desktop and mobile, and 'Extended Reality' (XR) displays (a catch-all terminology for augmented, virtual and mixed reality (AR/VR/MR) systems (Çöltekin et al., 2020)), to Dome and other surround systems. It encompasses human interaction ranging from graphical user interfaces to gestural 'natural' interfaces. Finally, these interfaces are involved in back-propagation loops, whereby queries are transmitted back to data repositories (e.g. selections of data), and metadata can be generated and logged (e.g. annotation of data).

Interspersed through the review that follows are details of the technical brief that emerges for this research from the given survey of available components.

3.1 Data

The practical implementation of an interactive data visualization pipeline is contingent on being able to work with the format in which the intended datasets are stored and accessed. This can be further enabled by some form of database, which, in general, is tailored to the needs of the particular scientific community. However, there are also considerations around the use of flexible and open formats and protocols to maximize interoperability.

File formats

Open scientific research employs open standards ranging from simple ASCII text files (e.g. .txt, .csv, .tsv) to sophisticated formats such as NetCDF (Rew and Davis, 1990) and HDF5 (Folk et al., 2011). These formats provide self-documenting machine-independent abstractions with embedded metadata, suitable for a wide variety of structured heterogeneous data. Consequently, they support a wide variety of query vectors and algorithmic operations employed by data mining and statistical pattern recognition

techniques. These include techniques for knowledge discovery in databases (KDD) (Fayyad et al., 1996; Fayyad and Smyth, 1999), presenting clear opportunities for visual analytics (Keim, 2001; Keim et al., 2010; Keim and Kriegel, 1996) and information visualization strategies (Ltifi et al., 2009).

Geosciences use a large number of specialist data formats and file types, which can lead to opaqueness about what the data represents, as well as data transfer problems between applications (Sen and Duffy, 2005). Self-describing formats are attractive precisely for this reason, where, for example, file header information can incorporate extractable metadata in a human-readable form. These may include characteristics such as what the data represents, where and how it was collected, similar to EXIF and IPTC specifications for photographic images (Tesic, 2005). It may also elucidate more esoteric features such as compression strategies and endian-ness, or byte order, which are especially important for binary formats. Efficiencies in file compression and handling can be achieved by the use of binary formats, noting that specific (often proprietary) software can be required to read the file. Binaries typically include bytes of information that describe some aspect of the structure of the file or the data contained, but in a less comprehensive manner than the self-documenting formats noted above.

Different file formats require a variety of data access methods, for example, the NCAR Command Language (NCL) (Brown, D et al., 2013) and NcML mark-up language (Nativi et al., 2004) for NetCDF. Similarly, ADIOS (Adapatable I/O System) (Liu et al., 2010) provides a metadata format for extremely large file sizes (e.g. petascale), utilizing an external XML file to describe user data (e.g. data types, sizes and I/O operations). ADIOS features a specialist binary format (BP) that handles IO staging (data IO) for HDF5 and NetCDF data. Together, agile data formats and flexible query languages facilitate storing, accessing and analyzing large geoscientific data repositories in new and useful ways.

One possible disjunction in a newly conceived data pipeline for interactive visualization of scientific data is the mismatch between SDIV formats and those employed by CIT. This requires the implementation of conversion mechanisms between formats, for instance from NetCDF (widely used in science) to an image file format (widely used by CIT) such as portable network graphics (PNG) or OpenEXR (Academy Software Foundation, 2019). Data conversions require appropriate parsing for data integrity and accurate representation, including, for instance, dimensionality, variables, range, and time variance. It should also be considered where and when this conversion occurs, for example, upon a server before the data is delivered over a network to a visualization client, or on the client system itself. Efficiencies can be delivered in both scenarios, where degrees of automation can be achieved using programmable conversion methods such as Python scripting. This indicates that some general domain knowledge about the data is required

at the outset of a visualization process.

Technical brief:

- Data formats are preferentially self-describing and open
- Data formats support sub-set selection (e.g. using a portion of the data)
- SDIV formats can be converted to other suitable, e.g. image, interchange formats for CIT

Databases

Data may be usefully held and managed in a database repository that might be generalized in form, or tailored to the scientific or other sub-discipline. Open databases facilitate new approaches to large datasets and computational analysis (Gray et al., 2005). General data users and scientific practices are moving away from 'flat' file systems stored on local hard drives (e.g. .csv), that are insufficiently agile and robust for many types of analytical enquiry, particularly relating to very large multidimensional and heterogeneous datasets. The general increase in data available to researchers is captured by the five major concerns of 'Big Data': data volume, data velocity, data variety, data veracity and data value (Sharma et al., 2014). These concern the amount of data, its speed of generation, heterogeneity, verifiability, and utility. This has driven the development of database hardware/software systems and file/query structures of greater versatility than flat file systems or relational (SQL) databases, e.g. BigTable, HBase, Cassandra, Hadoop, MapReduce, NoSQL and CrowdDB, amongst many others (Sharma et al., 2014). Different database structures provide a range of features pertinent to the types of data queries envisaged, the volume of data, and its means of delivery.

Technical brief:

- Database structures are preferentially self-describing and open
- Database supports a variety of query mechanisms (e.g. SQL, NoSQL)

3.2 Network

With very large or very changeable datasets it can be more efficient to move computation to the data rather than the data to the computation. The principal reason

behind this situation is the relatively poor rate of network bandwidth development vs the rate at which data is being collected (Emmott and Rison, 2008; Hey et al., 2009). Visualization of large datasets is intrinsically constrained by bandwidth: compression and down-sampling may be necessary if data are delivered over networks (Ahrens et al., 2009). Alternatives include cloud computing or implementing server-side subsampling via network transmissible query mechanisms (e.g. browser-based geographical area selection).

Servers

Networked cloud compute can take the form of virtual machines (VMs) located in immediate proximity to data repositories (Fox et al., 2009; Mell and Grance, 2010). VMs may be hosted on an infrastructure-as-a-service (IaaS) system such as NECTAR (Kirby, 2017), MASSIVE (MASSIVE Partners, 2020) and similar research infrastructure systems. VMs typically run Linux instances with configurable virtual hardware capabilities such as storage and memory, and a wide variety of software options. These include the ability to configure research-specific web-servers running, e.g. Apache (Apache Software Foundation, 1997) or nginx (F5 Inc., 2020), which are capable of hosting network accessible data services.

Protocols

For network data access, a THREDDS Data Server (TDS) is an attractive option for the storage and delivery of geoscientific data (Unidata UCAR, 2020). A TDS is an opensource data architecture developed for the geosciences. THREDDS (Thematic Real-time Environmental Distributed Data Services) are middleware, whose services, provided by the TDS, use a high-level data abstraction, the Common Data Model, enabling unified access to NetCDF and HDF5 data models (amongst others) via a common API, implemented in Java (Davis, 2002). Data is accessed via OPeNDAP ('Opensource Project for a Network Data Access Protocol'), a data transport architecture and protocol (Davis, 2002). The TDS can interpret OPeNDAP queries (programmatically conformed URLs passed to the server, parsed as queries) in order to request actions of the server upon its catalogue of data (an XML metadata repository). Actions may include retrieving subsets of given data (e.g. a date range, with a series of variables and parameters), virtual concatenations or other aggregations of datasets using NcML, NCL commands and a variety of NetCDF/HDF5 queries, such as slicing and dicing datacubes, and requesting vectors through data. THREDDS and OPeNDAP enable self-describing multidimensional data subsets to be accessed via shared compute resources. This has the advantage of by-passing network bottlenecks and minimizing the need to download the large datasets

typical of geosciences.

Technical brief:

- Data can be input, optionally, from a THREDDS server
- Data can be subsampled using OPeNDAP if necessary
- Data can be delivered, optionally, via a local filesystem
- Data can be converted between formats remotely or locally

3.3 Computer

Computer capabilities need to meet some minimum requirements for realtime visualization and interaction. Specifications vary contingent upon display and interaction expectations, for example, when considering the differing scenarios of mobile or desktop, XR or large-scale screen environments. In general, manufacturer specifications indicate the capabilities of devices, with increasing visual fidelity, data processing speed and interaction demands reflecting these. For very large display systems multiple computers will often be networked together to achieve the desired performance. This research aims to enrich the accessible toolset for visualization, and therefore targets a system that an individual researcher, or small research group, could realistically expect to access on a regular basis.

Specifications

Considerations include the capability to access network resources with reasonable speed (e.g. via high-speed ethernet or wifi), capacity to execute instructions rapidly (CPU speed), sufficient data storage and read/write performance (e.g. HDD or SSD), sufficient random access memory (RAM), multiple display support, USB connectivity and other features. These capabilities are met by recent-model high-performance desktop computers ('Pro' or high-end gaming models) and standard operating systems such as MacOS, Windows and Linux. Key performance bottlenecks include network speeds, bus speeds (e.g. PCI, USB, Thunderbolt), and GPU capabilities.

Technical brief:

- The computer used can be a performant standalone desktop system

GPU

The rapid evolution of data visualization is underpinned by significant increases in capability in computer graphics since the advent of the Graphics Processing Unit (GPU), a coin termed by NVIDIA in 1999 (McClanahan, 2010). It marks the first time in which the entire graphics pipeline (transform, lighting, triangulation, clipping, rendering (Hughes et al., 2013)) was implemented in hardware and in which the possibility of massively parallel compute first emerged (Owens et al., 2008). The GPU enabled the handling and processing of the large digital datasets beginning to emerge since the 1980s, as a result of cost-effective instrumentation, data storage and compute capacity (Menon and Hegde, 2015).

Graphics Processing Units (GPUs) are far more powerful than CPUs for certain types of tasks, such as parallel computing and computer graphics – for which they are specifically designed (Gregg and Hazelwood, 2011). The development of the GPU has largely been driven by consumer-demand for high-resolution interactive graphics in the multi-billion-dollar computer games industry (McClanahan, 2010). Interactive high-resolution data representations similarly demand graphics hardware of appropriate specification and leverage the exceptional development investment that has taken place in the CIT sector.

Technical brief:

- Graphics and user-interaction must be GPU-accelerated

3.4 Software

This section considers basic graphics rendering application programming interfaces (APIs) and graphics libraries. Scientific and dataflow programming are introduced, providing criteria for assessing the utility of various software categorized according to the SDIV/CIT schematic (Fig.1).

APIs

Graphics APIs define the graphics pipeline by which 2D and 3D data are processed on the GPU, before being drawn to screen, based upon fundamental specifications of geometric primitives, such as points, lines and polygons. In addition to geometry they can also specify depth-cues, antialiasing, shading models (including color, lighting and shadows), textures, motion blur and many other visual features, finally rasterizing these for screen display. The APIs provide a software interface to graphics hardware, using an open specification that supports every major operating system and the major hardware

vendors (NVIDIA, Intel, AMD, ARM). These include open APIs, OpenGL (Khronos Group, 2019) and Vulkan (Khronos Group, 2015), as well as proprietary APIs DirectX (Microsoft, 2020) and Metal (Apple Inc., 2020a). They support various shader languages for realtime rendering, such as the open standard GLSL (Khronos Group, 2020), proprietary Apple MSL (Apple Inc., 2020b) and Microsoft HLSL (Microsoft, 2019).

OpenGL, and its successor Vulkan, are cross-platform industry standards for real-time GPU-accelerated graphics rendering (Khronos Group, 2019, 2015). Opensource implementations are available through the Mesa3D graphics library (Mesa Developers, 2020). Apple deprecated OpenGL support in MacOS 10.14. Vulkan is supported on MacOS through the opensource MoltenVK API (KhronosGroup/MoltenVK, 2020) and Metal Vulkan SDK (LunarG Developers, 2020). With the forthcoming migration to Apple/ARM CPU/GPU architecture, future development for MacOS will optimally require employment of native Metal APIs.

Libraries

Built atop 2D and 3D graphics APIs are widely-used graphics libraries, which provide higher-level programmatic abstractions to the underlying graphics primitives, such as VTK (Kitware Inc., 2019), MatPlotLib (Hunter, 2007), Three.js (Mr.doob, 2019), D3.js (Bostock et al., 2011) and many more. Some are specialized for 2D graphics, such as MatPlotLib and D3.js, others for 3D, like VTK and Three.js. They use different back-ends with different rendering capabilities, notably with respect to GPU vs CPU usage and concomitant capabilities for realtime 2D and 3D interactivity, animation and data capacity (Rossant and Harris, 2013). JavaScript-based libraries are constrained by HTML5 DOM ('Document Object Model') context, such as SVG canvas objects, which restrict graphics to a 2D CPU-based render context, placing limitations on the complexity of scenes that can be drawn and animated for display. Similarly, WebGL, as a 'light-weight' subset of OpenGL, constrains 3D rendering and performance, though it is far more performant than HTML5 canvas elements (Halic et al., 2015).

Studies of web-based interactive graphics relative performance indicate that render speed is strongly influenced by software approaches employed (e.g. SVG, WebGL or OpenGL) as well as web-browser capabilities (Hoetzlein, 2012; Horak et al., 2018; Sarkissian, 2014; Yener, 2014). Of particular relevance is the 'interactivity limit', where framerates fall below 20fps (20Hz). Other studies of interactivity benchmark this at 10fps (Funkhouser and Séquin, 1993). Interaction lag is a serious bottleneck for usability: a 75ms lag measurably degrades human performance (MacKenzie and Ware, 1993; Ware and

Balakrishnan, 1994).

Technical brief:

- OpenGL/Vulkan are desirable for performant capabilities and open standards
- Aiming for minimum 20Hz screen refresh (20fps)
- Aiming for < 20ms lag

This precludes the use of synchronously network-delivered web-based display and interaction, deferring to asynchronous data access and loading to a local filesystem

Programming

Whilst some users engaged in geoscientific research are comfortable creating visual output using programming models and scripting (e.g. Python, R, Matlab), this does not simply translate to an interactive pipeline. Whilst these programmatic approaches have utility for some animated and other more advanced graphical displays, there are distinct limitations in interoperability with OpenGL and other graphics APIs (Campagnola et al., 2015; Rossant and Harris, 2013) that extend the design space beyond standard windowing systems and interfaces (Mei et al., 2018; Schulz et al., 2013; Sun et al., 2013). A number of SDIV and CIT software employ dataflow and visual programming in addition to conventional programming and scripting models. This can include capability for complex and adaptable UI design and implementation.

Dataflow/VPL

Research in visual programming languages (VPLs) has been undertaken throughout the history of computing (Boshernitsan and Downes, 2004; Burnett and Baker, 1994; Shu, 1986). VPLs allow users to create programs by graphically manipulating program elements rather than by specifying them textually (Noone and Mooney, 2018). This enables programmers to develop and iterate robust programs using simple drag-and-drop interfaces, where icons represent computational routines or objects. A variety of icon-based approaches exist, using a range of visual metaphors to express computational relations, functions, actions and objects (Idrees et al., 2018). Icons can be arranged upon a canvas and connected together by wires or pipes that indicate the flow of data between processing units, or as blocks that can be stacked in various configurations (Karvounidis et al., 2017; Mason and Dave, 2017). Icons feature ports which can accept incoming or output data of various types, or have associated parameters made available through

associated windows. VPLs are also commonly referred to as data flow programming, flow-based programming, and visual languages.

Pros	Cons
1] Ease of use for graphical applications	1] Lack of Standardization
2] Visual representation of algorithms	2] Over-simplification
3] Clarity of data flow	3] Visual clutter
4] Self-documenting	4] Inadequate documentation
5] Rapid-prototyping	5] Cross-platform limitations
6] Live coding/execution	6] Debugging problems
7] Compilation to code	7] Lack of code export

Table 1: Visual Programming Languages Pros and Cons

VPLs (Table 1) have well-known strengths and weaknesses (Burnett et al., 1995; Burnett and Baker, 1994). The VPL paradigm enables a fast-turn-around and easily comprehensible programming style that can be used to iterate design patterns, leading to rapid development and iteration of software prototypes. It makes software customization available to a wide range of users. VPL dataflow metaphors (Myers, 1990; Sanner et al., 2002; Sousa, 2012; Sutherland, 1966) are used in many current programming environments, including scientific ones. VPLs are notable for ease-of-use and logical transparency and are widely used in CIT GUI rapid prototyping applications, including for unconventional display and interaction IO.

Technical brief:

- Ability to rapidly iterate through UI design and implementation
- Ability to rapidly iterate through modes of user interaction

These are features found in dataflow VPLs deriving from the CIT domain.

Software Domains

This section lists relevant feature capabilities of representative visualization systems from the SDIV and CIT software domains. They are grouped according to the scheme (Fig.1) reflecting their domain expertise origin. Software is classified according to the following criteria:

Criterion	Key
Software Domain	SDIV, CIT
Graphics	Supports 2D, 3D, Volumetric (Vol)
OS	Cross-Platform incl. Linux (CP), Windows only (Win), MacOS only (MacOS)
VPL	Yes, No, or Scenegraph only (SG)
Language	Programming languages supported
API, Library, SL	Graphics API, Graphics libraries and Shader Language support: OpenGL, Vulkan (V), Metal (M), DirectX (DX)

	VTK, Pyqt, HTML5, WebGL, Canvas, SVG
Interactivity	Programmable (P) or built-in human-interaction device (HID) support
Network	Network support – Not Applicable (NA), Programmable (P), Yes (Y)

Table 3: Software categorization criteria

Software Domain	Graphics	OS	VPL	Language	API/Library/SL	Interactivity	Network
SDIV							
Bokeh	2D	CP	No	Python, R	HTML5, WebGL	P	P
Chaco	2D	CP	No	Python	Canvas	P	P
D3.js	2D/2.5D	CP	No	JS	HTML5 Canvas	P	P
Drishti	Vol	CP	No	C++	OpenGL	P	NA
Glue	2D/3D	CP	No	Python	OpenGL	P	P
Inviwo	3D/Vol	CP	Yes	C++, Python	OpenGL	P	NA
MatPlotLib	2D/3D	CP	No	Python	Python, SciPy	P	P
Mayavi	2D/3D/Vol	CP	No	Python	OpenGL	P	P
MeVisLab	2D/3D/Vol	CP	Yes	C++	OpenGL	P	Y
Orange	2D	CP	Yes	Python	Pyqt	P	Y
Paraview	2D/3D/Vol	CP	SG	Python, C++	OpenGL, VTK	P	Y
VisIt	2D/3D/Vol	CP	Yes	Python, C++	OpenGL	P	Y
VisPy	2D/3D/Vol	CP	No	Python/C++	OpenGL, GLSL	P	P
VTK	2D/3D/Vol	CP	SG	C++, Java, JS , Python, Tcl/Tk	OpenGL	P	NA
Workspace	2D/3D/Vol	CP	Yes	Python, C++	OpenGL	P	Y
Software Domain	Graphics	OS	VPL	Language	API/Library/SL	Interactivity	Network
CIT							
Blender	3D/Vol	CP	Yes	Python	OpenGL,M,V,DX, GLSL	P	P
Cinder	CC	CP	No	C++	OpenGL GLSL	P	P
Houdini	3D/Vol	CP	Yes	Python	OpenGL GLSL	P	P
Nodebox	2D	CP	Yes	Python, Clojure	Java OpenGL	No	P
OpenFrameworks	CC	CP	No	C++	OpenGL, GLSL	P	P
Pd (PureData)	2D/3D	CP	Yes	Python	OpenGL GLSL	HID	P
Processing	2D/3D	CP	No	Java	OpenGL, GLSL	P	P
Quartz Composer	2D/3D/Vol	MacOS	Yes	ObjectiveC, JS, OpenCL, Lua	CoreImage, GLSL, OpenGL/CL	HID	P
Touch Designer	2D/3D	Win/ MacOS	Yes	Python, C++	OpenGL, GLSL	HID	P
Vuo	2D/3D	MacOS	Yes	C, C++, ObjectiveC, JS, Python	OpenGL GLSL	HID	P
CIT: Game Engines							
Godot	2D/3D	CP	Yes	Python	OpenGL	HID	Y
Unity3D (2019.3)	2D/3D/Vol	CP	Yes (Bolt)	Python via Plugin, BOO, JS, C++	OpenGL,M,V, DX, WebGL, HLSL, Houdini Engine,	HID	Y
Unreal Engine (4.25)	2D/3D/Vol	CP	Yes (Blueprints)	Python via Plugin, BOO, JS, C++	OpenGL,M,V, DX, WebGL, Houdini Engine	HID	Y

Table 4: Software domain specifications

This research was undertaken on a MacPro 6,1, using OSX/MacOS systems 10.9 (Mavericks) – MacOS 10.15 (Catalina). At the commencement of this research, the stand-out development environment for rapid visualization prototyping, with GPU-based graphics acceleration, broad API support and stability, was provided by the native XCode development environment. XCode provides the Quartz Composer VPL (Apple Inc, 2007) for processing and rendering graphical data, building UIs and interaction modalities. It was deeply integrated into the operating system from OSX 10.4, providing a graphics-accelerated interface to underlying technologies such as OpenGL, GLSL, OpenCL, Core Image, Core Video, Core Animation and other OS subsystems. It is extensible via a plug-in architecture and permits the development of custom routines (patches/icons) in Objective-C, OpenCL, JavaScript and GLSL. This permits significant IO adaptability, including the incorporation of upstream and downstream data pre- and post-processing operations. QC programs can be distributed as freely-modifiable programs or can be compiled into stand-alone software applications via XCode.

The underlying architecture of software developed in a VPL provides a template for future development or deployment upon a different code base, as it can self-document and reveal underlying programmatic and logical structure. This informed the adoption of QC, as it can closely match structural and functional equivalents to common procedural dataflow architectures in CIT IDEs, such as computer game engines.

Technical brief:

- Support for OpenGL
- Support for GLSL shading language
- Support for 2D, 3D, Volumetric GPU-accelerated graphics
- Support for Javascript, C++, OpenCL
- Support for bespoke GUI implementation
- Support for HID (e.g. Mouse, Motion Controllers, Gestural Inputs)
- Desirable: support for stand-alone compiled program output

3.5 Interfaces

Geospatial visualization, display and analytic systems can take the form of conventional WIMP (Window-Icons-Menus-Pointers) stand-alone applications, web and database-connected solutions (Liu et al., 2014), as well as extended reality applications (XR/VR/AR/MR) (Van Krevelen and Poelman, 2010; Wilford, 1999) using novel HID

modalities (Bachmann et al., 2018; Klein et al., 2013; Morse et al., 2015). Rapid geoscientific visualization using an intuitive, interactive interface is not well represented in this software category. It presents novel opportunities for preliminary data reconnaissance, with potential extended functionality for screen technologies such as domes (Bourke, 2009; Li et al., 2014) and XR-type displays for immersive analytics (Cordeil et al., 2017; Marriott et al., 2018). Interface development methods need to be highly flexible in order to accommodate this wide repertoire of screen spaces, viewing scales and interaction modalities (Dimara and Perin, 2020). These must also include the ability to transmit queries back to datasets (e.g. through graphical data selection interfaces, or through transmitting commands to data servers) and to enable the generation of metadata through relevant file IO capability (e.g. saving logs of user interactions or user-generated annotations).

Screens

At a minimum, computers need to be capable of screen-refresh rates of at least 20 frames per second (20Hz) or more, for a user to perceive smooth interactive animation. Animation standards in the computer graphics industry mandate screen-refresh rates of 30-60Hz for monoscopic and 120Hz+ for stereoscopic presentations. High-resolution screens, e.g. WQHD (2560x1440px) or 4K (4096x2160), require many millions of pixels need to be drawn to screen every second. Current VR headsets, e.g. Vive Pro require 2880 x 1600px at 90Hz. Software, GPU and screen displays should be capable of 24-bit color.

UI & HID

In standard WIMP (Windows-Icons-Mouse-Pointer) interfaces, GUIs can take the form of buttons, sliders and fields that provide parametric feedback to the user, based upon standard human-interface guidelines articulated by design principles for standard operating systems. Current user-experience (UX) design builds upon research in graphical-user interfaces for HCI (Shneiderman et al., 2016). Whilst HCI research has led to usability and ergonomics standards (Bevan, 2001), UX and interaction design also considers cognitive aspects of user engagement (Benyon, 2014; Wania et al., 2006).

HCI also encompasses aspects of non-WIMP interaction, for example, touch interfaces on mobile devices; gestural interfaces that recognize hand movements, such as the Leap Motion Controller (Bachmann et al., 2018); wand, head and eye tracking in XR (Bednarz et al., 2019; Chandler et al., 2015; Fonnet and Prié, 2019), and other types of surround systems such as dome and CAVE displays (Kwasnitschka, 2017; Philips et al., 2015; Phillips, 2012). Different display systems require different types of input affordances and UI design implementation. Underpinning software architectures must be able to adapt and compensate for these different ways of seeing and interacting with onscreen visualizations, and propose methods whereby queries can be appropriately made of displayed data and how different types of metadata can be generated and recorded throughout the interaction with the system.

Technical brief:

- Support for WIMP and non-standard adaptable UI
- Support for performant standard desktop display
- Support for high-resolution displays
- Support for flexible display formats (e.g. XR, Vizwall, CAVE, Dome)
- Methods for defining and designing queries
- Methods for generating and recording metadata

3.6 Summary

The technical briefs derived from this survey of technologies, hardware and software, provided initial constraints on the development platform chosen for this research, operating at a point of intersection between SDIV and CIT. This involves balancing requirements for the scientifically accurate representation of data, with maximal flexibility in visualization display, interaction and UI design. Methods in SDIV have been developed to manage rigorously data repositories and network delivery of datasets, and a wide variety of sophisticated toolsets have arisen to enable precise forms of visualization. CIT provides methods for rapidly iterating prototype visualization designs and developing unconventional and adaptable interfaces and interactions. This expands the visual grammar of UI widgets and interaction modalities beyond the constraints of conventional check boxes, sliders and buttons, to include expanded modes of gestural interaction for XR, visual augmentations that can be driven by data, and large-scale cinematic presentation of data on very large immersive displays. Key bottlenecks in data transfer and data conversion between differing approaches have been identified, as have

limitations on the ability of different APIs and toolsets to accommodate specific visualization and interaction objectives beyond WIMP affordances.

The interactive visualization pipeline involves a complex chain of interconnected systems leading from data to display, to visual inspection, human interaction and interrogation. Visualization and interaction display systems are in a constant state of technological development, reflected, for instance, in rapid increase in sizes of datasets, GPU and CPU capabilities, screen resolutions, expansion of color gamuts, complexities of shaders and the capabilities of programming environments. Given this, it makes sense to utilize open methods that maintain maximal flexibility, including VPL approaches that may provide some protection against programmatic desuetude, by preserving high-level abstractions of software. At the center of this torrent of change is the human user, who remains relatively static in terms of perceptual and cognitive abilities. However, it is a measure of those abilities that technological systems for interactive visualization still have so much room for improvement.

The following three chapters of published research address the five key research aims through the interactive analysis of three types of geoscience datasets of increasing complexity and dimensionality. Chapter 4 develops a novel software methodology for the visualization of multi-parameter animated 1-D ocean storm time-series data, enabling the interactive visualization and annotation of multiple parameters in counterpoint. This articulates the principle of 'overview and detail' for large time-series data. The software architecture is developed and expanded in Chapter 5, with an application to representative 2D seismic data slices of the Australian continental lithosphere. This chapter accounts for the 'human in the loop' and develops the notion of well-posed visualization by incorporating aspects of human color perception in the optimization of applied color gradients. In Chapter 6, the architecture is extended to the interactive 3D volumetric visualization of a global tomographic dataset, with a focus upon the Indian Ocean region. This consolidates and expands upon findings from the previous two chapters, by exploiting animation of the dataset, precise color control and the visual faculties of stereopsis and depth perception. Finally, the usage and utility of an immersive visualization approach is introduced, by extending the display capabilities of the software architecture to interactive fulldome and extended-reality (XR) display systems. Features of these approaches and systems are discussed in detail in the relevant chapters and in the synthesis and discussion in Chapter 7, including current limitations and directions for future development. Chapter 8 provides a concise set of conclusions arising from these findings.

3.7 References

Abbasi, M., Vassilopoulou, P., Stergioulas, L., 2017. Technology roadmap for the Creative Industries. Creative Industries Journal 10, 40–58. https://doi.org/10/ggmq4h

Academy Software Foundation, 2019. OpenEXR [WWW Document]. URL https://www.openexr.com/ (accessed 1.21.20).

Ahrens, J.P., Woodring, J., DeMarle, D.E., Patchett, J., Maltrud, M., 2009. Interactive remote large-scale data visualization via prioritized multi-resolution streaming, in: Proceedings of the 2009 Workshop on Ultrascale Visualization - UltraVis '09. Presented at the 2009 Workshop, ACM Press, Portland, Oregon, pp. 1–10. https://doi.org/10.1145/1838544.1838545

Apache Software Foundation, 1997. Welcome! - The Apache HTTP Server Project [WWW Document]. URL https://httpd.apache.org/ (accessed 8.21.20).

Apple Inc., 2020a. Metal [WWW Document]. Apple Developer. URL https://developer.apple.com/metal/ (accessed 1.21.20).

Apple Inc., 2020b. Metal Shading Language Specification [WWW Document]. URL https://developer.apple.com/metal/Metal-Shading-Language-Specification.pdf#//apple_ref/doc/uid/TP40014364 (accessed 8.24.20).

Apple Inc, 2007. Quartz Composer User Guide [WWW Document]. URL https://developer.apple.com/library/mac/documentation/GraphicsImaging/Conceptual/QuartzComposerUserGuide/qc_intro/qc_intro.html (accessed 8.24.20).

Bachmann, D., Weichert, F., Rinkenauer, G., 2018. Review of Three-Dimensional Human-Computer Interaction with Focus on the Leap Motion Controller. Sensors 18, 2194. https://doi.org/10.3390/s18072194

Bae, J., Falkman, G., Helldin, T., Riveiro, M., 2019. Visual Data Analysis, in: Said, A., Torra, V. (Eds.), Data Science in Practice. Springer International Publishing, Cham, pp. 133–155. https://doi.org/10.1007/978-3-319-97556-6_8

Bednarz, T., Tobia, M., Nguyen, H., Branchaud, D., 2019. Immersive Analytics using Augmented Reality for Computational Fluid Dynamics Simulations, in: The 17th International Conference on Virtual-Reality Continuum and Its Applications in Industry on - VRCAI '19. Presented at the 17th International Conference on Virtual-Reality Continuum and its Applications in Industry, ACM Press, Brisbane, QLD, Australia, pp. 1–2. https://doi.org/10/ggjh3n

Benyon, D., 2014. Designing Interactive Systems: A comprehensive guide to HCI, UX and interaction design, 3rd Edition. ed. Pearson Education.

Bevan, N., 2001. International standards for HCI and usability. International Journal of Human-Computer Studies 55, 533–552. https://doi.org/10/d3ch5d

Boshernitsan, M., Downes, M.S., 2004. Visual programming languages: A survey (No. UCB/CSD-04-1368). Computer Science Division (EECS), University of California, Berkeley, California 94720.

Bostock, M., Ogievetsky, V., Heer, J., 2011. D^3 Data-Driven Documents. IEEE Transactions on Visualization and Computer Graphics 17, 2301–2309. https://doi.org/10.1109/TVCG.2011.185

Bourke, P., 2018. Scientific data visualisation using techniques normally reserved for more frivolous activities. GSTF Journal on Computing (JoC) 4.

Bourke, P., 2017. Digital Experiences.

Bourke, P., 2009. iDome: Immersive gaming with the Unity3D game engine. Computer Games and Allied Technology 9. https://doi.org/10/ggxxvb

Brodbeck, D., Mazza, R., Lalanne, D., 2009. Interactive Visualization - A Survey, in: Lalanne, D., Kohlas, J. (Eds.), Human Machine Interaction. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 27–46. https://doi.org/10.1007/978-3-642-00437-7_2

Brown, D, Brownrigg, R, Haley, M, Huang, W, 2013. The NCAR Command Language (NCL)(version 6.1.2). UCAR/NCAR Computational and Information Systems Laboratory, Boulder, CO.

Burnett, M.M., Baker, M.J., 1994. A classification system for visual programming languages. Journal of Visual Languages and Computing 5, 287–300. https://doi.org/10/fqwxc9

Burnett, M.M., Baker, M.J., Bohus, C., Carlson, P., Yang, S., Van Zee, P., 1995. Scaling up visual programming languages. Computer 28, 45–54. https://doi.org/10/dpp2h9

Campagnola, L., Klein, A., Larson, E., Rossant, C., Rougier, N.P., 2015. VisPy: harnessing the GPU for fast, high-level visualization, in: Proceedings of the 14th Python in Science Conference. https://doi.org/10.25080/Majora-7b98e3ed-00e

Chandler, T., Cordeil, M., Czauderna, T., Dwyer, T., Glowacki, J., Goncu, C., Klapperstueck, M., Klein, K., Marriott, K., Schreiber, F., Wilson, E., 2015. Immersive Analytics, in: 2015 Big Data Visual Analytics (BDVA). Presented at the 2015 Big Data Visual Analytics (BDVA), IEEE, Hobart, Australia, pp. 1–8. https://doi.org/10.1109/BDVA.2015.7314296

Chen, M., Hauser, H., Rheingans, P., Scheuermann, G. (Eds.), 2020. Foundations of Data Visualization. Springer International Publishing, Cham. https://doi.org/10.1007/978-3-030-34444-3

Çöltekin, A., Lochhead, I., Madden, M., Christophe, S., Devaux, A., Pettit, C., Lock, O., Shukla, S., Herman, L., Stachoň, Z., Kubíček, P., Snopková, D., Bernardes, S., Hedley, N., 2020. Extended Reality in Spatial Sciences: A Review of Research Challenges and Future Directions. ISPRS International Journal of Geo-Information 9, 439. https://doi.org/10.3390/ijgi9070439

Cordeil, M., Cunningham, A., Dwyer, T., Thomas, B.H., Marriott, K., 2017. ImAxes: Immersive Axes as Embodied Affordances for Interactive Multivariate Data Visualisation, in: Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology - UIST '17. Presented at the 30th Annual ACM Symposium, ACM Press, Québec City, QC, Canada, pp. 71–83. https://doi.org/10.1145/3126594.3126613

Davis, S., E.&. Caron, J.&. Domenico, B.&. Kambic, R.&. Nativi, 2002. 5.7 THREDDS: Publishing, Cataloging, Describing, and Discovering Scientific Datasets. UCAR/Unidata.

Dimara, E., Perin, C., 2020. What is Interaction for Data Visualization? IEEE Trans. Visual. Comput. Graphics 26, 119–129. https://doi.org/10/ggbwhz

Emmott, S., Rison, S., 2008. Towards 2020 science. Science in Parliament 65, 31–33.

F5 Inc., 2020. nginx [WWW Document]. URL https://nginx.org/en/ (accessed 8.21.20).

Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., 1996. From data mining to knowledge discovery in databases. AI magazine 17, 37.

Fayyad, U.M., Smyth, P., 1999. Cataloging and Mining Massive Datasets for Science Data Analysis. Journal of Computational and Graphical Statistics 8, 589. https://doi.org/10.2307/1390878

Folk, M., Heber, G., Koziol, Q., Pourmal, E., Robinson, D., 2011. An overview of the HDF5 technology suite and its applications, in: Proceedings of the EDBT/ICDT 2011 Workshop on Array Databases - AD '11. Presented at the EDBT/ICDT 2011 Workshop, ACM Press, Uppsala, Sweden, pp. 36–47. https://doi.org/10.1145/1966895.1966900

Fonnet, A., Prié, Y., 2019. Survey of Immersive Analytics. IEEE Transactions on Visualization and Computer Graphics 1–1. https://doi.org/10/gf73gm

Fox, A., Griffith, R., Joseph, A., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., 2009. Above the clouds: A Berkeley view of cloud computing. Dept. Electrical Eng. and Comput. Sciences, University of California, Berkeley, Rep. UCB/EECS 28, 13.

Funkhouser, T.A., Séquin, C.H., 1993. Adaptive display algorithm for interactive frame rates during visualization of complex virtual environments, in: Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '93. Presented at the 20th annual conference, ACM Press, Not Known, pp. 247–254. https://doi.org/10.1145/166117.166149

Gray, J., Liu, D.T., Nieto-Santisteban, M., Szalay, A., DeWitt, D.J., Heber, G., 2005. Scientific data management in the coming decade. ACM SIGMOD Record 34, 34–41. https://doi.org/10.1145/1107499.1107503

Gregg, C., Hazelwood, K., 2011. Where is the data? Why you cannot debate CPU vs. GPU performance without the answer, in: (IEEE ISPASS) IEEE International Symposium on Performance Analysis of Systems and Software. Presented at the IEEE International Symposium on Performance Analysis of Systems and Software, IEEE, Austin, TX, USA, pp. 134–144. https://doi.org/10.1109/ISPASS.2011.5762730

Halic, T., Ahn, W., De, S., 2015. Optimization model for web based multimodal interactive simulations. Expert Systems with Applications 42, 5245–5255. https://doi.org/10.1016/j.eswa.2015.02.026

Hartley, J. (Ed.), 2005. Creative industries. Blackwell Pub, Malden, MA.

Hey, T., Tansley, S., Tolle, K.M., others, 2009. The fourth paradigm: data-intensive scientific discovery. Microsoft research Redmond, WA.

Hoetzlein, R.C., 2012. Graphics Performance in Rich Internet Applications. IEEE Computer Graphics and Applications 32, 98–104. https://doi.org/10.1109/MCG.2012.102

Horak, T., Kister, U., Dachselt, R., 2018. Comparing Rendering Performance of Common Web Technologies for Large Graphs, in: Poster Program of the 2018 IEEE VIS Conference, VIS. p. 2.

Hughes, J.F., Van Dam, A., Foley, J.D., McGuire, M., Feiner, S.K., Sklar, D.F., Akeley, K., 2013. Computer graphics: principles and practice. Addison-Wesley Professional.

Hunter, J.D., 2007. Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering 9, 90–95. https://doi.org/10.1109/MCSE.2007.55

Idrees, M., Aslam, F., Shahzad, K., Sarwar, S.M., 2018. Towards a universal framework for visual programming languages. Pak. J. Engg. Appl. Sci. Vol 55–65.

Karvounidis, T., Argyriou, I., Ladias, A., Douligeris, C., 2017. A design and evaluation framework for visual programming codes, in: 2017 IEEE Global Engineering Education Conference (EDUCON). Presented at the 2017 IEEE Global Engineering Education Conference (EDUCON), pp. 999–1007. https://doi.org/10/ggc99k

Keim, D., 2001. Visual exploration of large data sets, in: Communications of the ACM. pp. 38–44. https://doi.org/10.1145/381641.381656

Keim, D., Kriegel, H.-P., 1996. Visualization techniques for mining large databases: a comparison. IEEE Transactions on Knowledge and Data Engineering 8, 923–938. https://doi.org/10.1109/69.553159

Keim, D., Mansmann, F., Thomas, J., 2010. Visual analytics: how much visualization and how much analytics? ACM SIGKDD Explorations Newsletter 11, 5. https://doi.org/10.1145/1809400.1809403

Kenderdine, S., 2010. Immersive Visualization Architectures and Situated Embodiments of Culture and Heritage, in: 2010 14th International Conference Information Visualisation. Presented at the 2010 14th International Conference Information Visualisation (IV), IEEE, London, TBD, United Kingdom, pp. 408–414. https://doi.org/10.1109/IV.2010.63

Khronos Group, 2020. OpenGL Shading Language - OpenGL Wiki [WWW Document]. URL https://www.khronos.org/opengl/wiki/OpenGL_Shading_Language (accessed 8.22.20).

Khronos Group, 2019. OpenGL - The Industry Standard for High Performance Graphics [WWW Document]. URL https://opengl.org/ (accessed 1.21.20).

Khronos Group, 2015. Vulkan - Industry Forged [WWW Document]. The Khronos Group. URL https://www.khronos.org/vulkan/ (accessed 8.22.20).

KhronosGroup/MoltenVK, 2020. KhronosGroup/MoltenVK [WWW Document]. URL https://github.com/KhronosGroup/MoltenVK (accessed 8.22.20).

Kirby, J., 2017. NeCTAR-Australian Research Cloud.

Kitware Inc., 2019. VTK - The Visualization Toolkit [WWW Document]. VTK. URL https://vtk.org/ (accessed 7.2.19).

Klein, J., Reuling, D., Grimm, J., Pfau, A., Lefloch, D., Lambers, M., Kolb, A., 2013. User Interface for Volume Rendering in Virtual Reality Environments. arXiv:1302.2024 [cs].

Kwasnitschka, T., 2017. Planetariums — not just for kids. Nature 544, 395–395. https://doi.org/10/gf4ksh

Lewi, H., Smith, W., Lehn, D. vom, Cooke, S., 2019. The Routledge International Handbook of New Digital Practices in Galleries, Libraries, Archives, Museums and Heritage Sites. Routledge.

Li, S., Huang, Y., Tri, V.-S., Elvek, J., Wan, S., Kjallstrom, J., Andersson, N., Johansson, M., Lejerskar, D., 2014. Interactive theater-sized dome design for edutainment and immersive training, in:

Proceedings of the 2014 Virtual Reality International Conference. ACM, p. 8. https://doi.org/10/gf8z7r

Liu, Q., Klasky, S., Podhorszki, N., Lofstead, J., Abbasi, H., Chang, C.S., Cummings, J., Dinakar, D., Docan, C., Ethier, S., Grout, R., Kordenbrock, T., Lin, Z., Ma, X., Oldfield, R., Parashar, M., Romosan, A., Samatova, N., Schwan, K., Shoshani, A., Tian, Y., Wolf, M., Yu, W., Zhang, F., Zheng, F., 2010. ADIOS: Powering I/O to extreme scale computing, in: SciDAC 2010. Presented at the SciDAC, Oak Ridge National Laboratory, Chattanooga, Tennessee, p. 6.

Liu, S., Cui, W., Wu, Y., Liu, M., 2014. A survey on information visualization: recent advances and challenges. The Visual Computer 30, 1373–1393. https://doi.org/10.1007/s00371-013-0892-3

Ltifi, H., Ayed, M.B., Alimi, A.M., Lepreux, S., 2009. Survey of information visualization techniques for exploitation in KDD, in: 2009 IEEE/ACS International Conference on Computer Systems and Applications. Presented at the 2009 IEEE/ACS International Conference on Computer Systems and Applications, pp. 218–225. https://doi.org/10/c7wphz

LunarG Developers, 2020. LunarXchange [WWW Document]. URL https://vulkan.lunarg.com/sdk/home (accessed 8.22.20).

MacKenzie, I.S., Ware, C., 1993. Lag as a determinant of human performance in interactive systems, in: Proceedings of the INTERACT'93 and CHI'93 Conference on Human Factors in Computing Systems. ACM, pp. 488–493. https://doi.org/10/dr7rj6

Marriott, K., Chen, J., Hlawatsch, M., Itoh, T., Nacenta, M.A., Reina, G., Stuerzlinger, W., 2018. Immersive Analytics: Time to Reconsider the Value of 3D for Information Visualisation, in: Marriott, K., Schreiber, F., Dwyer, T., Klein, K., Riche, N.H., Itoh, T., Stuerzlinger, W., Thomas, B.H. (Eds.), Immersive Analytics, Lecture Notes in Computer Science (LNCS. Springer International Publishing, pp. 25–55. https://doi.org/10.1007/978-3-030-01388-2_2

Mason, D., Dave, K., 2017. Block-based versus flow-based programming for naive programmers, in: 2017 IEEE Blocks and Beyond Workshop (B B). Presented at the 2017 IEEE Blocks and Beyond Workshop (B B), pp. 25–28. https://doi.org/10/ggc99j

MASSIVE Partners, 2020. MASSIVE - about [WWW Document]. URL https://www.massive.org.au/about/ (accessed 8.21.20).

McClanahan, C., 2010. History and Evolution of GPU Architecture. A Paper Survey http://mcclanahoochie. com/blog/wpcontent/uploads/2011/03/gpu-hist-paper. pdf.

Mei, H., Ma, Y., Wei, Y., Chen, W., 2018. The design space of construction tools for information visualization: A survey. Journal of Visual Languages & Computing 44, 120–132. https://doi.org/10/gdhs5z

Mell, P., Grance, T., 2010. The NIST definition of cloud computing. Communications of the ACM 53, 50.

Menon, S.P., Hegde, N.P., 2015. A survey of tools and applications in big data, in: 2015 IEEE 9th International Conference on Intelligent Systems and Control (ISCO). Presented at the 2015 IEEE 9th International Conference on Intelligent Systems and Control (ISCO), IEEE, Coimbatore, India, pp. 1–7. https://doi.org/10.1109/ISCO.2015.7282364

Mesa Developers, 2020. Home | The Mesa 3D Graphics Library [WWW Document]. URL https://mesa3d.org/ (accessed 8.22.20).

Microsoft, 2020. DirectX programming - UWP applications [WWW Document]. URL https://docs.microsoft.com/en-us/windows/uwp/gaming/directx-programming (accessed 8.22.20).

Microsoft, 2019. Programming Guide for HLSL - Win32 apps [WWW Document]. URL https://docs.microsoft.com/en-us/windows/win32/direct3dhlsl/dx-graphics-hlsl-pguide (accessed 8.22.20).

Morse, P., Reading, A., Lueg, C., Kenderdine, S., 2015. TaggerVR: Interactive Data Analytics for Geoscience - A Novel Interface for Interactive Visual Analytics of Large Geoscientific Datasets in Cloud Repositories, in: 2015 Big Data Visual Analytics (BDVA). Presented at the 2015 Big Data Visual Analytics (BDVA), IEEE, Hobart, Australia, pp. 1–2. https://doi.org/10.1109/BDVA.2015.7314303

Mr.doob, 2019. Three.js JavaScript 3D library. [WWW Document]. URL https://github.com/mrdoob/three.js (accessed 9.2.20).

Munzner, T., 2014. Visualization Analysis and Design. A K Peters/CRC Press. https://doi.org/10.1201/b17511

Myers, B.A., 1990. Taxonomies of visual programming and program visualization. Journal of Visual Languages & Computing 1, 97–123. https://doi.org/10.1016/S1045-926X(05)80036-9

Nativi, S., Caron, J., Domenico, B., 2004. NcML-G_ML: Encoding NetCDF Datasets Using GML., in: DEXA Workshops. pp. 804–808. https://doi.org/10.1109/DEXA.2004.1333574

Noone, M., Mooney, A., 2018. Visual and textual programming languages: a systematic review of the literature. Journal of Computers in Education 5, 149–174. https://doi.org/10/ggc95c

Owens, J.D., Houston, M., Luebke, D., Green, S., Stone, J.E., Phillips, J.C., 2008. GPU Computing. Proceedings of the IEEE 96, 879–899. https://doi.org/10.1109/JPROC.2008.917757

Philips, A., Walz, A., Bergner, A., Graeff, T., Heistermann, M., Kienzler, S., Korup, O., Lipp, T., Schwanghart, W., Zeilinger, G., 2015. Immersive 3D geovisualization in higher education. Journal of Geography in Higher Education 39, 437–449. https://doi.org/10.1080/03098265.2015.1066314

Phillips, M., 2012. There is no dome? Digital Creativity 23, 48–57. https://doi.org/10.1080/14626268.2012.666252

Purchase, H.C., Andrienko, N., Jankun-Kelly, T.J., Ward, M., 2008. Theoretical Foundations of Information Visualization, in: Kerren, A., Stasko, J.T., Fekete, J.-D., North, C. (Eds.), Information Visualization. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 46–64. https://doi.org/10.1007/978-3-540-70956-5_3

Rew, R., Davis, G., 1990. NetCDF: an interface for scientific data access. IEEE Computer Graphics and Applications 10, 76–82. https://doi.org/10.1109/38.56302

Rossant, C., Harris, K.D., 2013. Hardware-accelerated interactive data visualization for neuroscience in Python. Frontiers in Neuroinformatics 7. https://doi.org/10.3389/fninf.2013.00036

Sanner, M.F., Stoffler, D., Olson, A.J., 2002. ViPEr, a visual programming environment for Python, in: Proceedings of the 10th International Python Conference. pp. 103–115.

Sarkissian, G., 2014. Evaluation of HTML5 Graphics for Data Structure Visualization (M.Sc.). California State University, Northridge.

Schulz, H., Nocke, T., Heitzler, M., Schumann, H., 2013. A Design Space of Visualization Tasks. IEEE Transactions on Visualization and Computer Graphics 19, 2366–2375. https://doi.org/10.1109/TVCG.2013.120

Sen, M., Duffy, T., 2005. GeoSciML: Development of a generic GeoScience Markup Language. Computers & Geosciences 31, 1095–1103. https://doi.org/10/cq72fj

Sharma, Sugam, Tim, U.S., Wong, J., Gadia, S., Sharma, Subhash, 2014. A Brief Review on Leading Big Data Models. Data Science Journal 13, 138–157. https://doi.org/10.2481/dsj.14-041

Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., Diakopoulos, N., 2016. Designing the User Interface: Strategies for Effective Human-Computer Interaction, 6th ed. Pearson.

Shu, N.C., 1986. Visual programming languages: A perspective and a dimensional analysis, in: Visual Languages. Springer, pp. 11–34. https://doi.org/10/d227sn

Sousa, T.B., 2012. Dataflow programming concept, languages and applications, in: Doctoral Symposium on Informatics Engineering. pp. 323–334.

Sun, G.-D., Wu, Y.-C., Liang, R.-H., Liu, S.-X., 2013. A Survey of Visual Analytics Techniques and Applications: State-of-the-Art Research and Future Challenges. Journal of Computer Science and Technology 28, 852–867. https://doi.org/10.1007/s11390-013-1383-8

Sutherland, W.R., 1966. The on-line graphical specification of computer procedures. (PhD Thesis). Massachusetts Institute of Technology.

Telea, A.C., 2015. Data visualization: principles and practice. CRC Press.

Tesic, J., 2005. Metadata practices for consumer photos. IEEE MultiMedia 12, 86–92. https://doi.org/10/fmhmvz

Unidata UCAR, 2020. Unidata | THREDDS Data Server (TDS) [WWW Document]. URL https://www.unidata.ucar.edu/software/tds/ (accessed 8.21.20).

Van Krevelen, D.W.F., Poelman, R., 2010. A survey of augmented reality technologies, applications and limitations. International journal of virtual reality 9, 1–20. https://doi.org/10/ggxxt5

Wania, C.E., Atwood, M.E., McCain, K.W., 2006. How do design and evaluation interrelate in HCI research?, in: Proceedings of the 6th ACM Conference on Designing Interactive Systems - DIS '06. Presented at the 6th ACM conference, ACM Press, University Park, PA, USA, p. 90. https://doi.org/10.1145/1142405.1142421

Ward, M.O., Grinstein, G., Keim, D., 2010. Interactive data visualization: foundations, techniques, and applications. AK Peters/CRC Press, Natick, Massachusetts.

Ware, C., 2013. Information visualization: perception for design, 3rd ed. Elsevier, Amsterdam, Holland.

Ware, C., Balakrishnan, R., 1994. Reaching for objects in VR displays: lag and frame rate. ACM Transactions on Computer-Human Interaction 1, 331–356. https://doi.org/10.1145/198425.198426

Wilford, J., 1999. Scientific visualisation and 3D modelling applications for mineral exploration and environmental management. AGSO Research Newsletter 31.

Yener, Y., 2014. Hardwarebeschleunigtes Rendern im Browser mit WebGL (M.Sc.). Fachochschule Köln, Cologne, Germany.

Chapter 4 : Animated analysis of geoscientific datasets: An interactive graphical application

4 Publication details

This chapter is published as :

Morse, P., Reading, A., Lueg, C., 2017. Animated analysis of geoscientific datasets: An interactive graphical application. Computers & Geosciences 109, 87–94. https://doi.org/10/gcmxvw

Note that thesis page range 64 – 71, are enumerated in journal format text as pp.87-94.

Supplemental material referenced by this publication is appended at the end of this document in the Supplements section.

Online Supplemental materials (including software, movies, images) are available at:

https://doi.org/10.5281/zenodo.3264037

Chapter 4: https://github.com/pemorse/data-visualization-tools

Computers and Geosciences 109 (2017) 87–94

Contents lists available at ScienceDirect

Computers and Geosciences

journal homepage: www.elsevier.com/locate/cageo

Research paper

Animated analysis of geoscientific datasets: An interactive graphical application

Peter Morse [a,*], Anya Reading [a], Christopher Lueg [b]

[a] School of Physical Sciences (Earth Sciences), University of Tasmania, Hobart, TAS, 7001, Australia
[b] School of Engineering and ICT, University of Tasmania, Hobart, TAS, 7001, Australia

ARTICLE INFO

Keywords:
Visual analytics
Interactive
Animated
Time series analysis

ABSTRACT

Geoscientists are required to analyze and draw conclusions from increasingly large volumes of data. There is a need to recognise and characterise features and changing patterns of Earth observables within such large datasets. It is also necessary to identify significant subsets of the data for more detailed analysis.

We present an innovative, interactive software tool and workflow to visualise, characterise, sample and tag large geoscientific datasets from both local and cloud-based repositories. It uses an animated interface and human-computer interaction to utilise the capacity of human expert observers to identify features via enhanced visual analytics. 'Tagger' enables users to analyze datasets that are too large in volume to be drawn legibly on a reasonable number of single static plots. Users interact with the moving graphical display, tagging data ranges of interest for subsequent attention. The tool provides a rapid pre-pass process using fast GPU-based OpenGL graphics and data-handling and is coded in the Quartz Composer visual programing language (VPL) on Mac OSX. It makes use of interoperable data formats, and cloud-based (or local) data storage and compute.

In a case study, Tagger was used to characterise a decade (2000–2009) of data recorded by the Cape Sorell Waverider Buoy, located approximately 10 km off the west coast of Tasmania, Australia. These data serve as a proxy for the understanding of Southern Ocean storminess, which has both local and global implications. This example shows use of the tool to identify and characterise 4 different types of storm and non-storm events during this time. Events characterised in this way are compared with conventional analysis, noting advantages and limitations of data analysis using animation and human interaction. Tagger provides a new ability to make use of humans as feature detectors in computer-based analysis of large-volume geosciences and other data.

1. Introduction

1.1. Geosciences data

The volume of data generated by scientific instruments, sensor systems and computational models is growing at an increasing pace (Hey et al., 2009) and most geosciences disciplines and technologies exhibit this trend (Sellars et al., 2013). Typical datasets might comprise time variant observations at fixed geographic locations such as tide-gauges and oceanographic buoy data, spatially distributed data such as soil geochemistry analyses, or combined spatial/time variant data such as those observed by satellites. Other large datasets include outputs of model simulations and forecasts. Interpretations of geosciences data are commonly carried out using graphs and maps, however, there is a necessary limitation on how much information can be presented in a single representation and how many graphs or maps can be usefully incorporated in one study (Ware, 2012; Munzner, 2014). Statistical analyses afford the ability to summarise in a systematic way but at the expense of exploratory analysis and much pattern characterization.

Graphical and other diagrammatic representations of data can be thought of as visualisations (Tufte, 1990, 1997; Few, 2009). Existing visualisation tools are generally used for the presentation of results rather than being an inherent part of the data inference process (Ware, 2012; Few, 2015; Victor, 2005). However, interactive visual analytics could be used much more widely at earlier stages in the workflow, prospectively generating new insight into the underlying data (Thomas and Cook, 2005; Keim et al., 2008; Ward et al., 2010).

* Corresponding author.
E-mail address: peter.morse@utas.edu.au (P. Morse).

http://dx.doi.org/10.1016/j.cageo.2017.07.006
Received 2 September 2016; Received in revised form 24 July 2017; Accepted 29 July 2017
Available online 7 August 2017

1.2. Human interaction with data displays

The human comprehension of graphical representations is complex, as it involves the interplay between physiology, visual perception, cognition and experience, together with factors such as information density, dataset size, graphic design and color (Ware, 2012; Healey and Enns, 2012). Some flexibility in the display allows for physiological and experiential differences in scientific data analysts. Expert human analysts can identify structures and patterns that are challenging for statistical classification systems. Both comprehension and pattern detection are influenced by the graphical abstraction of data and the levels of detail present (Carpenter and Shah, 1998; Friel et al., 2001). Taken together, there is a point where visual data density impairs comprehensibility (Walker et al., 2016), as illustrated by the two static plots (Fig. 1).

1.3. Static and animated interactive representations of data

Geoscientists and other data scientists routinely use software packages to visualise and present results as static graphs and charts. This graphing and underpinning 'spreadsheet' document model is user-friendly, robust and has been in use for many decades (Campbell-Kelly, 2003; ch.12) with more recent capabilities that enable dynamic updating of variables. Whilst adequate for many scientific purposes, this approach becomes unwieldy and can lead to missed insights, as datasets increase in size and complexity. As an alternative, many users engaged in scientific research are comfortable creating visual output using programming models and scripting (e.g. Python, R, Matlab) as datasets become larger.

Animated interactive representations of data enable more sophisticated visualisations for interpretation. With the advent of high performance computer graphics, the repertoire of graph types has significantly expanded, including adaptive and network representations (Herman et al., 2000; Liu et al., 2014). Programming languages widely used in science such as Python are also moving beyond the conventional static charts and visualisations produced by MatPlotLib (Hunter, 2007), with an array of tools that leverage programming languages in concert with vector graphics libraries (Khronos Group, 2016), e.g. Bokeh (Continuum Analytics, 2015). Alternatively, using a programming language together with OpenGL enables the processing of significantly greater data volumes (e.g. VTK and Paraview (Kitware, 2016); Vispy (VisPy Developers, 2015)). These softwares combine, in varying degrees, features for interaction and animation, constrained by their underlying architecture (Kloss, 2009). 2D plotting libraries predominantly execute upon the CPU, limiting their ability to handle very large datasets and real-time interactivity. In contrast, OpenGL-based plotting systems (Supplement 1) afford the possibility of handling millions to billions of datapoints as well as handling high-frame-rate animation and interactivity, via the transaction of events and data across CPU and GPU (Rossant and Harris, 2013).

1.4. GPU-enhanced scientific-oriented software

Scientific visualisation has been able to exploit these hardware developments for the analysis and visualisation of massive datasets (Guha et al., 2005; Cuntz et al., 2007; Fogal et al., 2010). Examples of previous studies in the geosciences include the development of sophisticated computer simulations, such as weather forecasting (Rautenhaus et al.,

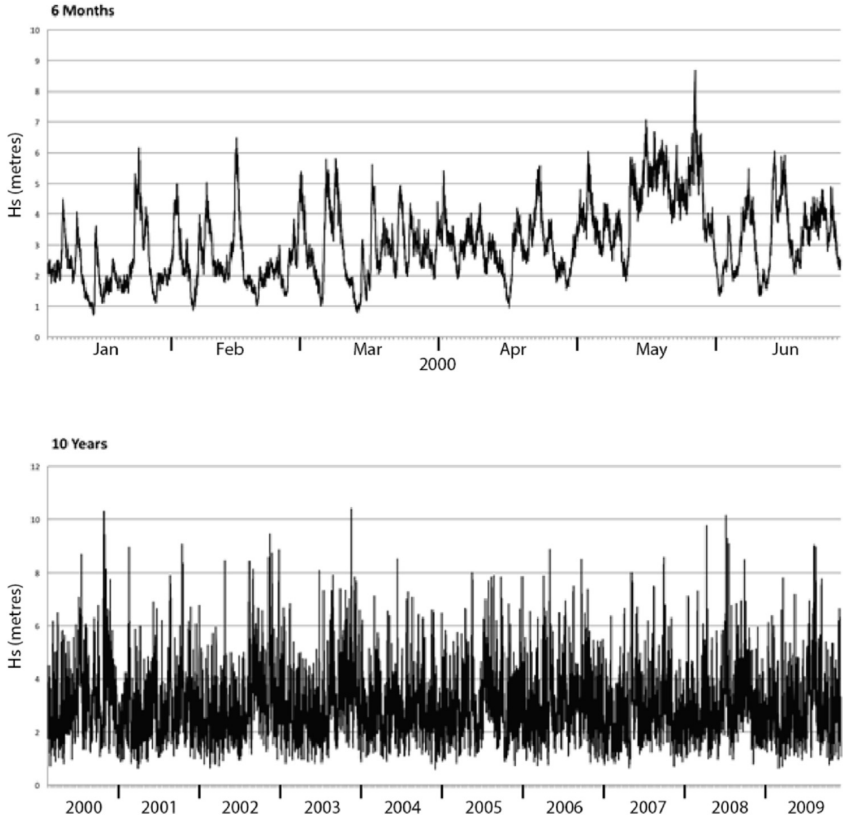


Fig. 1. Significant wave height (Hs) over time, recorded at the Cape Sorell Waverider Buoy, West of Tasmania, Australia: Upper plot: 6 months (1/1/2000-30/6/2000) (8583 Samples), Lower plot 10 years (2000–2009) (165,484 Samples).

2015) and climate modelling (Michalakes and Vachharajani, 2008) and other notable contributions making use of GPUs from cosmology (Kähler et al., 2006; Sainio, 2010; Bard et al., 2013). More general packages with sophisticated client-server models that implement distributed back-end compute and rendering have also been developed, e.g. Paraview (Henderson et al., 2004), VisIt (Ahern et al., 2013), SciVi (Ryabinin and Chuprina, 2013). While recognizing the utility of these packages, their potential for customization of the visual interactive interface is constrained by their underlying architecture and implementation of interface controls.

We make use of the practical convergence between the technical capabilities afforded by GPU-enabled graphics and processing (Supplement 1 provides an extensive technical background), and the goals of contemporary scientific data analysis. The interaction functionality and the capabilities afforded by performant GPU-accelerated graphics are combined with cloud-enabled ability to connect with agile scientific data formats. We use the NeCTAR Research Cloud (National Research Infrastructure for Australia, 2016) to host a virtual machine configured with a THREDDS data server (TDS) (Supplement 2). Given the desirability of exposing data to the scientific user in a dynamic way, we present a new software application for animated, interactive data analysis.

2. Tagger: interactive graphical application

2.1. Application aims and development framework

We address the need for a highly performant visual analytics platform that connects with scientific data formats, enables the reconnaissance viewing of large datasets, and has an interactive selection/tagging capability. The new application Tagger aims to leverage the capabilities of the human visual-cognitive system to observe patterns in data, through an animated interface, that may lead to new insight into the underlying patterns or other characteristics of the data. The application aims to be readily customizable for different data throughflow scenarios and to be easily adaptable for a variety of human-computer interfaces, providing the insightful use of visual enhancements to aid scientific data characterization, pattern detection and data selection for further analysis.

Tagger is programmed in Quartz Composer (QC), a Visual Programming Language (VPL) delivered as part of Apple's freely available Xcode development environment (Apple Computer, 2007). As QC is not widely used for programming in geosciences, a short overview is provided as Supplement 3. QC provides a high-level VPL interface to underlying technologies such as OpenGL, OpenCL, Core Image, Core Video and other OSX graphics technologies. It is extensible via a plug-in architecture and also permits the development of text-coded routines in Objective-C, OpenCL or Javascript, in concert with the visual programming paradigm. Quartz Compositions (data flow programs) can be distributed as freely-modifiable programs or can be compiled into stand-alone software applications via Xcode. Of key relevance to the functionality of this language for scientific data visualisation is the ease of customising inputs, outputs and workflows. The visual programming environment enables rapid iteration through application prototypes to working interfaces for a wider spectrum of users than would be possible with a non-visual language (Sousa, 2012).

Tagger is a multi-layered QC program constructed in a modular fashion, comprising a series of routines for performing manipulations upon the data streaming in and being rendered to screen. The application accesses data through a cloud-hosted server accessible via OPeNDAP and THREDDS (see Supplement 1 and 2). It can also read local.txt and .csv data files.

2.2. Tagger user interface

The Tagger GUI is arranged in a series of panes, menus and tools (Fig. 2): a) Application Interface Settings (drop down menu); b) The Data Graphing pane; c) Selection and Tagging Tool, d) Meta Data Display; e) Data Visualisation Controls; f) Data Input and Output Controls.

The Data Graphing pane can be switched between a variety of graph types and views (e.g. independent overlay, stacked fraction, stacked value, various tiling modes). The usual mode of operation is that the data are animated, in the chosen graph type/view. The graph moves across the screen for perusal by the analyst, data selection and tagging. A robust link to the input scientific data is made through the Meta Data Display which shows a moving list of all data being visualised as well as dynamically generated summaries, e.g. the total number of fields, data point values and current range selection. The choice of data to be

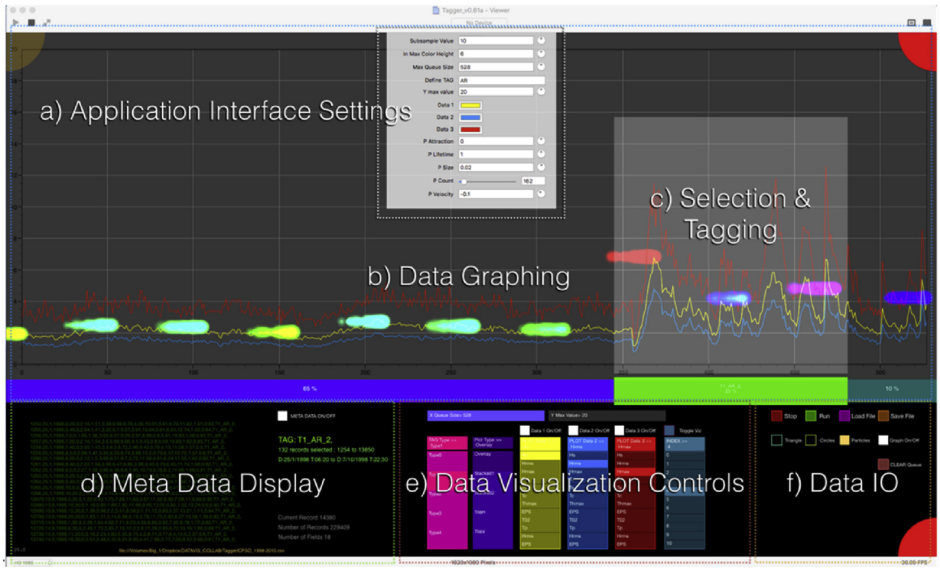


Fig. 2. Tagger GUI (v0.61a).

visualised within the chosen file is made through the Data Visualisation Controls pane. File selection, either local or cloud-based, is made through the Data IO pane toggle buttons and via standard operating system file open and save windows. Toggle buttons are placed here to start and stop animation, as well as toggling on and off visualisation overlays. Application Interface Settings (drop down menu) enable further customization to suit user experience and the nature of the incoming data.

Using the application for data selection and tagging should be intuitive to scientific users. Areas of the graph that displays events of interest are selected using the mouse, with the 't' key being pressed to tag that data. This can be done whether the data are 'playing' or stopped. The data and its tag are written out to a local saved file, each tag is automatically numerically incremented and appended to that file as the analysis proceeds. Tags are user-defined alphanumeric strings that can be used as keys to identify different types of events.

The Data IO pane can be used to assign visual analytic aids, such as particle systems, to the on-screen data divisions, overlaying the graph. Particle systems are small graphics objects displayed on screen, designed to draw attention to a particular feature of the data, such as the animated data stream exceeding a threshold level. The graph can be turned off and the data visualised as an abstract animation. This potentially accelerates movement through the data and the procedure of tagging by the user. Examples of different usage modes are given in our illustrative case study below and in the accompanying video (Supplement 6).

2.3. Tagger application capabilities

The application achieves the objectives as visualisation software that can perform data queries and visual analytics tasks on both local and remote Earth Sciences data. Cloud storage and a VM THREDDS data server are implemented (Supplement 2), hosting an OPeNDAP network-addressable database of NetCDF files.

The application capabilities enable the user to:

(i) Graph time-variant data using a performant, animated interface to visualise an extensive stream of time-variant data.
(ii) Access and query both local and remote multi-parametric data held in variety of formats including agile (NetCDF) and traditional (.txt, csv) data formats.
(iii) Zoom in and out on the datastream in order to examine it at different timescales and levels of detail.
(iv) Overlay unconventional visual cues driven by visually enhanced animation on the animated datastream, in order to alert users to different types of events occurring in the data.
(v) Counterpoise multiple data streams in the animation so that different values can be compared.
(vi) Pause the animation, make a visual selection of some of the data currently in the main window, and tag regions of interest, for output to a local saved file.

The user-defined selections append a user-defined metadata tag to the lines corresponding to that selection. The local file 'captures' the expert observation of the user, who has some familiarity with the type of data being observed; these saved observational and event subsets can later be subjected to conventional computational (e.g. parametric) analysis. A subset of the application capabilities is illustrated in the following case study.

3. Illustrative case study: Cape Sorell Waverider Buoy data

The Cape Sorell Waverider Buoy (WRB) is a swell-measuring buoy, located approximately 10 km west of Macquarie Harbor, Tasmania (42.9S, 145.03E) (Australian Bureau of Meteorology, 2016). It was chosen as an indicator location for the variability of large wave events on Australia's southern margin (Macaulay, 2009), where it is openly exposed to the westerly wind-driven Southern Ocean wave climate. Data from this buoy may be interpreted as a proxy for marine storms in the Australian sector of the Southern Ocean (Hemer, 2010). The aim of this case study analysis was to provide an appraisal of storm character in addition to, and complementing, a parametric approach (Hemer, 2010; Hemer et al., 2010).

3.1. Data and methods

The data used in our study represent wave height information recorded continuously over 10 years (2000–2009), sampling every 30 min, captured in 18 data fields. This totals over 4 million (4,099,680) data points recorded across 227,760 rows. Whilst this is an order of magnitude less than data files considered in science as high volume, it is nevertheless a challenge to visualise *interactively* this quantity of data in a way that is scientifically useful. The case study is readily scalable to higher data volumes (Supplement 7) and provides an example in which dynamical visualisation can afford a reconnaissance of characteristics of major features of the data.

An observer (with broad general science experience) undertook a visual analysis of 10 years of Cape Sorell Waverider Buoy data (2000–2009). The aim of the reconnaissance was to identify and characterise storm events, and other features of the wave activity measured at the buoy. Three data streams were visualised, Maximum (Hmax), Significant (Hs) and RMS (Hrms) Wave Heights. Tagger was configured to display 11 days of data across the graph. The WRB samples 48 times every 24hrs, generating 480 samples per 10 days. Half-day 'visual buffers' were chosen on either side of this data, resulting in a display in which 528 samples moved across the graph. This provided an adequate amount of data to display, for analysis purposes, as well as a usable frame rate for display and animation (20 frames per second).

Enhanced visualisation was applied to the middle data stream (Hs) such that storms, unusually quiet interludes and other features were easily identified. A particle-system overlay was chosen, generating streams of particles at each division (0–10) across the display – 11 points in total. Particle dimensions and colorization are parametrically driven by underlying values in the data. Other features of particles, such as scaling, velocity and lifespan, were user-adjusted in the dropdown application interface settings, until an initial set of useful visual parameters were arrived at: RGB color-space mapping identifies low value inputs as yellow-green, medians as cyan-blue, high values as magenta-red.

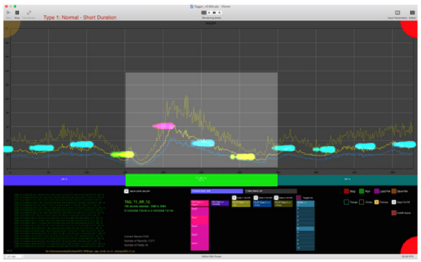
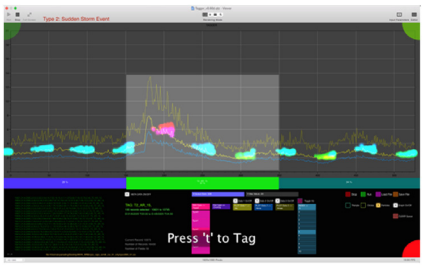
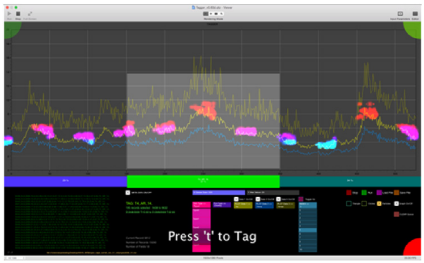
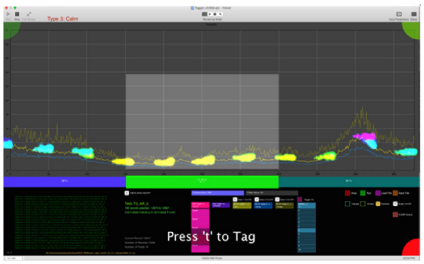
3.2. Results

The visualisation of wave height information using the Tagger application was run on a Mac Pro 2014 model, with 12GB VRAM on two ATI graphics cards. This proved more than sufficient to run the software at a highly-performant frame rate. After some initial trials, visual analysis for the 10-year dataset took 5 h (approximately 25 min per year of data, with a 10 min break each hour). The trials served the dual functions of running the data to identify a logical visual classification, and gaining facility with the application.

Storm events were identified with an arbitrary threshold of Hs > 6 m. They range from single to multi-day duration, each with distinctive characteristics. The most obvious features of the visual characterization were tagged for subsequent analysis (Table 1). Observations fell into four categories: normal storms, sudden storms, long storms and calm periods (T1-T4) with other remarkable features being tagged as T0.

After some experimentation, we chose to assist the visualisation through the use of coloured particles, e.g. in the 'Normal' storm', T1, example (Table 1) background levels of activity are median prior to and after the storm (blue particles); immediately prior to the storm and afterwards we see a significant lull in activity (yellow particles). Magenta or red visualisation enhancements indicate storm level activity.

Table 1
Visual characterisations of storm events made for a decade of Waverider Buoy data from Cape Sorell. The visualisation screen shots show enhanced visualisation using particle streams. The pale centre rectangle indicates the data subset that is captured and written to file with each 'Tag', for subsequent analysis.

Tag and Classification	
T1 – Normal storm Onset: relatively gradual, over 6–12 h. Duration: 36–48 h. Decline: over 24–48 h or longer.	
T2 – Sudden storm Onset: rapid, 3 h or less. Duration: 24–36 h. Decline: over a 24–48 h.	
T3 – Long storm Onset: relatively gradual, over 6–12 h. Duration: greater than 4 days. Decline: over 24–48 h or longer.	
T4 – Calm Onset: n/a Duration: 4 days or longer. Decline: n/a	

3.3. Interpretation

A timeline of the storm events identified is shown (Fig. 3) for the 10-year dataset including the different categories of storm identified. The new insights obtained through the animated visual analysis are: 1) characterization of different storm categories and 2) identification of other events (e.g. calm periods). One category, the 'sudden storm' could be subsequently detected through an automated algorithm, but this category would not have been evident without reconnaissance viewing and the benefit of hindsight. After the second or third viewing, more wave character events become evident, e.g. lengthy elevated wave activity, a 'substorm' that falls below the level of a full storm (Fig. 4). Hence, there is also considerable potential for other observations that are more difficult to categorise.

We compare a conventional analysis (Hemer, 2010) counting 'exceedances' with the animated visual analysis (this case study) total storm counts (Fig. 5). The total counts are comparable with a shift relating to the chosen threshold levels. Our study, however, adds the ability to characterise qualitatively the storms, and also to show other useful characteristics of the wave climate, i.e. the propensity for long calms, and long 'substorms' which may be of utility in assessing the potential for marine operations or wave power initiatives. While the value of the event characterization may have to be experienced by the analyst to be fully appreciated, it is evident (i.e. from Table 1) that there is great value in being able to add enhanced information to conventional analyses and further explore features of the data.

4. Discussion

4.1. New capacity in data analysis

Foremost in our criteria regarding the appraisal of animated analysis, and the Tagger application, is whether the visualisation reconnaissance process makes a positive difference in analysis and knowledge generation. The full potential of this analysis will be clear only to those who have run a full dataset 2 or 3 times, however, we have been convinced by our experience of the new insights that emerge from such an approach. As evidenced through use of the new application, animated analysis has proven successful in making possible 1) the visual reconnaissance analysis of a large dataset and 2) the identification of previously unrecognised events. Feature characteristics and captured metadata could potentially be translated into algorithmic constraints for ongoing automated analysis (Fayyad and Smyth, 1999; Amershi et al., 2014; Hammer et al., 2014).

One strong feature is the ability to make a key connection between modelled scenarios (e.g. Hemer et al., 2010) and the experienced result of such scenarios (e.g. different storm types). This has significance in the translation of research to practical outcomes for industry and community. A further strong feature is the ability to connect spot observations to models of wide scope. Where on-ground observations exist over a suitable time frame, the application will facilitate feedback into general models with the potential for more effective ground-truthing of such models. It is also worth emphasizing the added value that comes from being able to view efficiently a long run of data: features emerge that would not otherwise be identified by standard means.

4.2. Limitations

Tagger is alpha-stage research software. It is robust (having never crashed during our case study) but could undergo further performance optimization. It requires a fast computer with a high-end GPU and 8GB + RAM to run smoothly, and Mac OSX 10.10 + with a run-time environment that includes XCode Tools and QC plugins. We anticipate porting the VPL architecture of Tagger from QC to Vuo (Vuo Developers, 2016) or Origami Studio (Facebook Design, 2016), analogous VPLs developed in response to the uncertain support and development of QC by Apple. Vuo and Origami Studio will have near feature-parity with QC, modern faster implementations of OpenGL, as well as cross-platform support.

A short-coming of the interactive analysis is eventual user fatigue, with the risk of human errors affecting the tagged file. We recommend a few minutes break every half hour to a maximum of about 5 h in one day. A related remark is that human interaction is dependent upon the decisions and consistency of the end-user. Hence, Tagger is intended as a tool for first-pass visualisation of data, for annotating and developing a sense of what is occurring in the data. With practice, as with any tool, we anticipate that repeat runs will inform the development of insight. Tagged data can be subjected to quantitative analysis, informed by the visual reconnaissance, as part of the scientist's analytical toolkit.

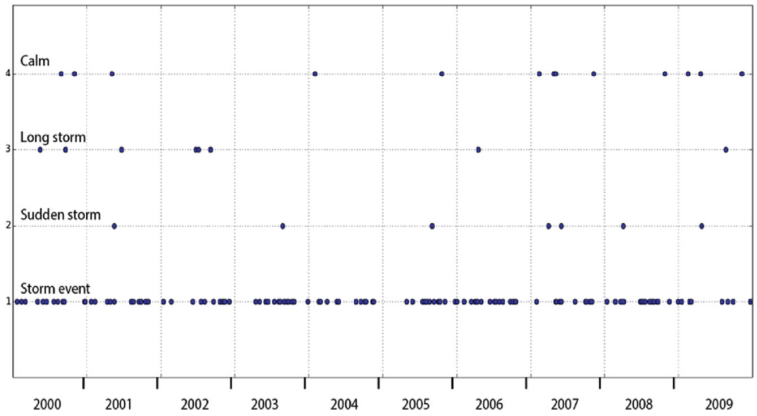


Fig. 3. Summary of storm event types identified during the animated analysis of a decade (2000–2009) of wave height data from the Cape Sorell Waverider Buoy.

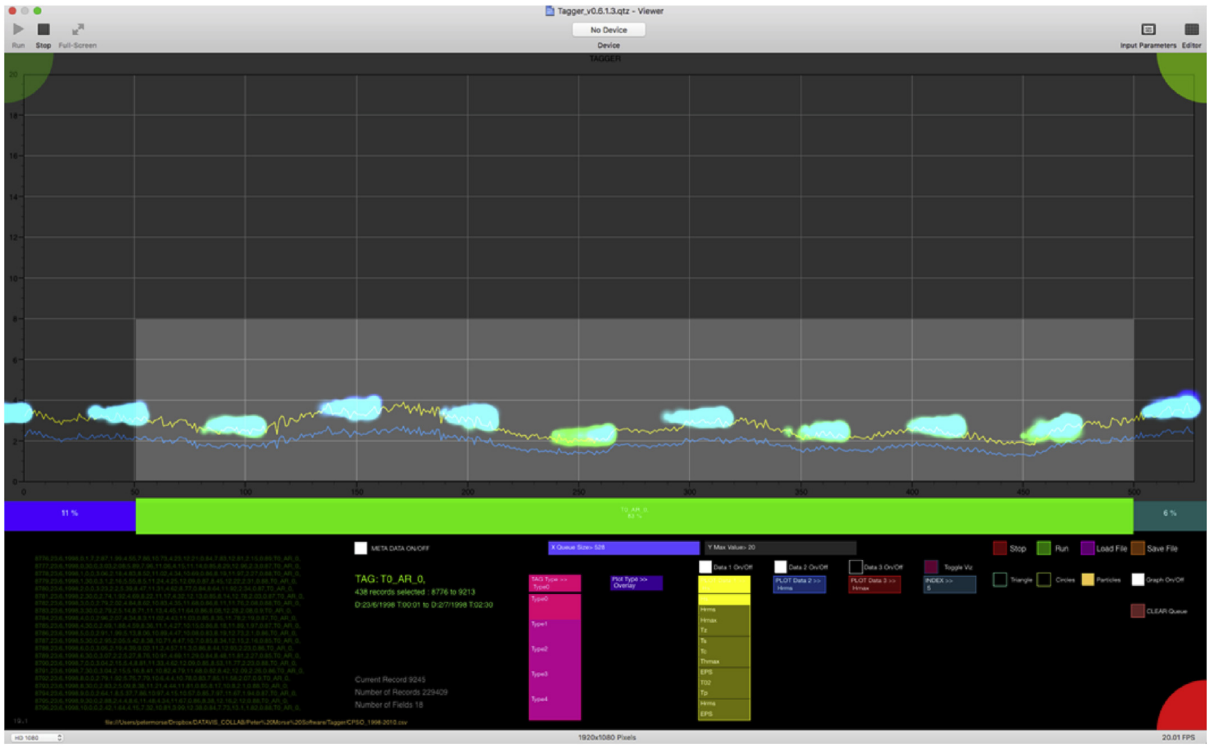


Fig. 4. 'Substorm' – blue particles indicate elevated activity over a prolonged period. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

4.3. Flexible application

Tagger can be used to analyze visually a wide variety of numerical data, represented as 1-D value series, optionally, multiple 1-D series. Examples of these include other types of buoy or time variant sensor data, borehole depth-relative data and spatial transects. The data should be formatted as a local. csv/.txt or remote NetCDF file, with appropriate OPeNDAP query structures. Tagger can be configured to ingest data over a network, for human expert analysis.

Quartz Composer is extremely versatile in its output beyond the conventional 2D screen display interface used in the case study (Supplement 3). The Tagger GUI has been developed in various trial versions to demonstrate the utility of a 3D GUI, with multiple graphs displayed in an XYZ coordinate space. Its outputs can be arbitrarily mapped to other types of scenarios, such as immersive dome projection (e.g. fulldome display), projection mapping (multiple projectors in arbitrary arrangement) and for augmented reality (AR) and virtual reality headsets (e.g. Oculus Rift)(Morse et al., 2015). It can also accept inputs from a variety of interface devices, extending the modalities of user interaction beyond conventional mouse/keyboard input.

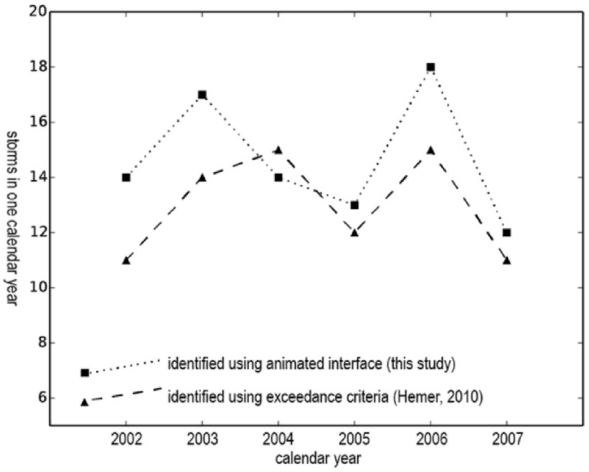


Fig. 5. Storms in calendar years 2002–2007 identified using the animated interface (square symbols, dotted line) compared with storms identified using numerical exceedance criteria. The association between the two is in general very good. The year with the greatest data loss is 2004 (Supplement 5) and is affected by differences in the way that data loss is handled by each method.

5. Conclusions

We have demonstrated the utility of a highly performant OpenGL GPU-based software tool for the animated analysis of time-series geosciences data. The software has successfully interrogated a cloud-based data repository running a THREDDS data server, delivering requested data subsets of NetCDF files via OPeNDAP queries. Features include the ability to customise visual enhancements to aid feature identification by a human observer. Data throughflow may also be varied to match the desired analysis task and optimise human interaction. As a demonstration case study, our software has been used to characterise qualitatively a variety of different storm and non-storm events in Waverider Buoy data in a manner that complements conventional statistical analysis or static graphing.

Increasing volumes and complexities of data in the earth sciences demand new ways of interrogating the data, including the development of innovative forms of human-computer interaction that can facilitate discovery of underlying characteristics and patterns. The ability to interface highly-performant GPU-based OpenGL software with a cloud-hosted TDS system has broad applicability. Tagger extends the capability of visual analytics in a geosciences context.

Acknowledgements

PM holds a Tasmanian Graduate Research Scholarship from the University of Tasmania. Cape Sorell Waverider Buoy data are recorded and made available through a joint initiative Australian Bureau of Meterology/CSIRO Marine and Atmospheric Research initiative. The authors thank Mark Hemer for his expertise regarding the Cape Sorell Waverider Buoy data used in the case study, and thank Daniel Keim and Stephen Hardy for their insights from conference presentations and other discussions. The manuscript has been improved by constructive comments from two anonymous reviewers.

Tagger is available on Github: https://github.com/pemorse/Tagger.

Appendix A. Supplementary data

Supplementary data related to this article can be found at http://dx.doi.org/10.1016/j.cageo.2017.07.006.

References

Ahern, S., Brugger, E., Whitlock, B., Meredith, J.S., Biagas, K., Miller, M.C., Childs, H., 2013. VisIt Experiences with sustainable software. arXiv Prepr. arXiv:1309.1796.
Amershi, S., Cakmak, M., Knox, W.B., Kulesza, T., 2014. "Power to the people: the role of humans in interactive machine learning. AI Mag. 35 (no. 4), 105–120.
Apple Computer, 2007. Quartz Composer User Guide. https://developer.apple.com/library/mac/documentation/GraphicsImaging/Conceptual/QuartzComposerUserGuide/qc_intro/qc_intro.html. (Accessed 25 August 2016).
Australian Bureau of Meteorology, 2016. Cape Sorell Waverider Buoy Observations. http://www.bom.gov.au/products/IDT65015.shtml. (Accessed 27 January 2016).
Bard, D., Bellis, M., Allen, M.T., Yepremyan, H., Kratochvil, J.M., 2013. Cosmological calculations on the GPU. Astronomy Comput. 1, 17–22.
Campbell-Kelly, M., 2003. The History of Mathematical Tables: from Sumer to Spreadsheets. Oxford University Press.
Carpenter, P.A., Shah, P., 1998. A model of the perceptual and conceptual processes in graph comprehension. J. Exp. Psychol. Appl. 4 (no. 2), 75.
Continuum Analytics, 2015. Bokeh Python Library. http://bokeh.pydata.org. (Accessed 19 January 2016).
Cuntz, N., Kolb, A., Leidl, M., Rezk-Salama, C., Böttinger, M., 2007. GPU-based dynamic flow visualization for climate research applications. SimVis 371–384.
Facebook Design, 2016. Origami Studio. https://facebook.github.io/origami/. (Accessed 22 August 2016).
Fayyad, U.M., Smyth, P., 1999. Cataloging and mining massive datasets for science data analysis. J. Comput. Graph. Statistics 8 (no. 3), 589–610.
Few, S., 2009. Now You See it: Simple Visualization Techniques for Quantitative Analysis. Analytics Press.
Few, S., 2015. Signal: Understanding what Matters in a World of Noise. Analytics Press.
Fogal, T., Childs, H., Shankar, S., Krüger, J., Bergeron, R.D., Hatcher, P., 2010. Large data visualization on distributed memory multi-GPU clusters. In: Proceedings of the Conference on High Performance Graphics, pp. 57–66.
Friel, S.N., Curcio, F.R., Bright, G.W., 2001. Making sense of graphs: critical factors influencing comprehension and instructional implications. J. Res. Math. Educ. 124–158.
Guha, S., Krisnan, S., Venkatasubramanian, S., 2005. Data visualization and mining using the gpu. In: Tutorial at ACM SIGKDD, vol. 5.
Hammer, B., He, H., Martinetz, T., 2014. Learning and modelling big data. In: ESANN 2014 Proceedings.
Healey, C.G., Enns, J.T., 2012. Attention and visual memory in visualization and computer graphics. IEEE Trans. Vis. Comput. Graph. 18 (no. 7), 1170–1188.
Hemer, M.A., 2010. Historical trends in Southern Ocean storminess: long-term variability of extreme wave heights at Cape Sorell, Tasmania. Geophys. Res. Lett. 37 (no. 18).
Hemer, M.A., Church, J.A., Hunter, J.R., 2010. Variability and trends in the directional wave climate of the Southern Hemisphere. Int. J. Climatol. 30 (no. 4), 475–491.
Henderson, A., Ahrens, J., Law, C., et al., 2004. The ParaView Guide. Kitware, Clifton Park, NY.
Herman, I., Melançon, G., Marshall, M.S., 2000. Graph visualization and navigation in information visualization: a survey. Vis. Comput. Graph. IEEE Trans. 6 (no. 1), 24–43.
Hey, A.J., Tansley, S., Tolle, K.M., et al., 2009. The Fourth Paradigm: Data-intensive Scientific Discovery. Microsoft Research, Redmond, WA.
Hunter, J.D., 2007. Matplotlib: a 2D graphics environment. Comput. Sci. Eng. 9 (no. 3), 90–95.
Kähler, R., Wise, J., Abel, T., Hege, H.-C., 2006. GPU-assisted raycasting for cosmological adaptive mesh refinement simulations. In: Machiraju, R., Möller, T. (Eds.), Eurographics/IEEE VGTC Workshop on Volume Graphics (Boston, Massachusetts, USA, 2006). Eurographics Association, pp. 103–110.
Keim, D., Andrienko, G., Fekete, J.-D., Görg, C., Kohlhammer, J., Melançon, G., 2008. Visual Analytics: Definition, Process, and Challenges. Springer.
Khronos Group, 2016. OpenVG. https://www.khronos.org/openvg/. (Accessed 27 January 2016).
Kitware, 2016. Visualization Toolkit (VTK). http://www.vtk.org. (Accessed 27 January 2016).
Kloss, G.K., 2009. Python data plotting and visualisation extravaganza. In: Proceedings of the First Kiwi PyCon (New Zealand). Christchurch: the Python Papers Monograph.
Liu, S., Cui, W., Wu, Y., Liu, M., 2014. A survey on information visualization: recent advances and challenges. Vis. Comput. 30 (no. 12), 1373–1393.
Macaulay, C., 2009. Tracking bigger wave action. ECOS Mag.
Michalakes, J., Vachharajani, M., 2008. GPU acceleration of numerical weather prediction. Parallel Process. Lett. 18 (no. 04), 531–548.
Morse, P., Reading, A., Lueg, C., Kenderdine, S., 2015. TaggerVR: interactive data analytics for geoscience-a novel interface for interactive visual analytics of large geoscientific datasets in cloud repositories. In: Big Data Visual Analytics (BDVA), 2015, pp. 1–2.
Munzner, T., 2014. Visualization Analysis and Design. CRC Press.
National Research Infrastructure for Australia, 2016. NeCTAR Research Cloud. https://www.nectar.org.au. (Accessed 27 January 2016).
Rautenhaus, M., Kern, M., Schäfler, A., Westermann, R., 2015. "Three-dimensional visualization of ensemble weather forecasts-Part 1: the visualization tool Met. 3D (version 1.0). Geosci. Model Dev. 8 (no. 7), 2329–2353.
Rossant, C., Harris, K.D., 2013. "Hardware-accelerated interactive data visualization for neuroscience in Python. Front. Neuroinformatics 7.
Ryabinin, K., Chuprina, S., 2013. "Adaptive scientific visualization system for desktop computers and mobile devices. Procedia Comput. Sci. 18, 722–731.

Sainio, J., 2010. CUDAEASY-a GPU accelerated cosmological lattice program. Comput. Phys. Commun. 181 (no. 5), 906–912.
Sellars, S., Nguyen, P., Chu, W., Gao, X., Hsu, K., Sorooshian, S., 2013. Computational earth science: big data transformed into insight. Eos, Trans. Am. Geophys. Union 94 (no. 32), 277–278.
Sousa, T.B., 2012. Dataflow programming concept, languages and applications. In: Doctoral Symposium on Informatics Engineering.
Thomas, J.J., Cook, K.A., 2005. Illuminating the Path: the Research and Development Agenda for Visual Analytics. IEEE Computer Society Press.
Tufte, E.R., 1990. Envisioning Information. Graphics Press.
Tufte, E.R., 1997. Visual Explanations: Images and Quantities, Evidence and Narrative. Graphics Press, Cheshire, CT.
Victor, B., 2005. Magic Ink: Information Software and the Graphical Interface, 2005. http://worrydream.com/MagicInk/. (Accessed 28 April 2016).
VisPy Developers, 2015. Vispy Python Library. http://vispy.org/. (Accessed 19 January 2016).
Vuo Developers, 2016. Vuo. https://vuo.org. (Accessed 27 January 2016).
Walker, J., Borgo, R., Jones, M.W., 2016. TimeNotes: a study on effective chart visualization and interaction techniques for time-series data. Vis. Comput. Graph. IEEE Trans. 22 (no. 1), 549–558.
Ward, M.O., Grinstein, G., Keim, D., 2010. Interactive Data Visualization: Foundations, Techniques, and Applications. CRC Press.
Ware, C., 2012. Information Visualization: Perception for Design. Elsevier.

Chapter 5: Well-Posed Geoscientific Visualization Through Interactive Color Mapping

5 Publication details

This chapter is published as:

Morse, P.E., Reading, A.M., Stål, T., 2019. Well-Posed Geoscientific Visualization Through Interactive Color Mapping. Front. Earth Sci. 7, 274. https://doi.org/10/ggbjzq

Note that thesis page range 73 – 89, are enumerated in journal format text as pp.1-17.

Online Supplemental materials (including software and high-resolution images) are available at:

https://doi.org/10.5281/zenodo.3264037

https://github.com/pemorse/data-visualization-tools

frontiers in Earth Science

TECHNOLOGY AND CODE
published: 22 October 2019
doi: 10.3389/feart.2019.00274

Well-Posed Geoscientific Visualization Through Interactive Color Mapping

Peter E. Morse[1]*, *Anya M. Reading*[1,2] *and Tobias Stål*[1,2]

[1] School of Natural Sciences (Earth Sciences), University of Tasmania, Hobart, TAS, Australia, [2] Institute for Marine and Antarctic Studies, University of Tasmania, Hobart, TAS, Australia

Scientific visualization aims to present numerical values, or categorical information, in a way that enables the researcher to make an inference that furthers knowledge. Well-posed visualizations need to consider the characteristics of the data, the display environment, and human visual capacity. In the geosciences, visualizations are commonly applied to spatially varying continuous information or results. In this contribution we make use of a suite of newly written computer applications which enable spatially varying data to be displayed in a performant graphics environment. We present a comparison of color-mapping using illustrative color spaces (RGB, CIELAB). The interactive applications display the gradient paths through the chosen color spaces. This facilitates the creation of color-maps that accommodate the non-uniformity of human color perception, producing an image where genuine features are seen. We also take account of aspects of a dataset such as parameter uncertainty. For an illustrative case study using a seismic tomography result, we find that the use of RGB color-mapping can introduce non-linearities in the visualization, potentially leading to incorrect inference. Interpolation in CIELAB color space enables the creation of perceptually uniform linear gradients that match the underlying data, along with a simply computable metric for color difference, ΔE. This color space assists accuracy and reproducibility of visualization results. Well-posed scientific visualization requires both "visual literacy" and "visual numeracy" on an equal footing with clearly written text. It is anticipated that this current work, with the included color-maps and software, will lead to wider usage of informed color-mapping in the geosciences.

Keywords: data visualization, seismic tomography, feature identification, color mapping, color space, CIELab, RGB

OPEN ACCESS

Edited by:
Christine Thomas,
University of Münster, Germany

Reviewed by:
Fabio Crameri,
University of Oslo, Norway
Thomas Lecocq,
Royal Observatory of Belgium, Belgium

***Correspondence:**
Peter E. Morse
peter.morse@utas.edu.au

Specialty section:
This article was submitted to Solid Earth Geophysics, a section of the journal Frontiers in Earth Science

Received: *18 July 2019*
Accepted: *04 October 2019*
Published: *22 October 2019*

Citation:
Morse PE, Reading AM and Stål T (2019) Well-Posed Geoscientific Visualization Through Interactive Color Mapping. Front. Earth Sci. 7:274. doi: 10.3389/feart.2019.00274

INTRODUCTION

Graphical representations in the form of static diagrams, plots, and charts form a fundamental part of the scientific toolset. Scientific visualization aims to reveal and explore relationships in data and assist in the development of robust inference, posing two initial questions of data: "*Is what we see really there?*" and "*Is there something there we cannot see?*" The first question encapsulates the interplay between scientific curiosity and *apophenia*—"the innate human ability to see pattern in noise" (Wickham et al., 2010; Cook, 2017). The second question exposes the concept of "missed discovery," where the analyst is unaware that unperceived structures await discovery (Buja et al., 2009).

Important challenges are thus posed to software designed for visual analytics (Keim et al., 2006) and data-based graphical inference (Cook et al., 2016). Interactive computer-based visualizations can expand the explanatory and exploratory capabilities of scientific software. A significant body of research in visualization, interactivity, analysis and design (e.g., Tukey, 1990; Wilkinson, 2005; Ward et al., 2010; Ware, 2013; Munzner, 2014) provides the foundations for visualization practice.

Well-posed visualizations clearly elicit features of underlying data values, maintaining an overt awareness of the risks of representational ambiguity and error (Rougier et al., 2014). Given the constraints of a human-computer visualization system (Haber and McNabb, 1990; Hansen and Johnson, 2005), they can reveal structures and patterns that may be elusive to other, e.g., statistical, approaches (Tukey, 1977, 1990; Tufte, 1990). The informational capacity of static images can be extended by incorporating elements of interactivity (Ward et al., 2010).

Interactivity enables the exploration of the design-space for visualization (Schulz et al., 2013), including constraints for visual encoding and interaction idioms (Munzner, 2014), creating a feedback loop between user and visualization system. Representations may be examined in detail, forming an important part of the analytical and inferential processes (Keim, 2001; Keim et al., 2006) that actively facilitate conceptual model building and analyses (Keim et al., 2010; Ward et al., 2010; Harold et al., 2016).

BACKGROUND AND RELATED WORK

The Interactive Visualization Process

A model of the interactive visualization process (**Figure 1**) proposes three principal components: "Data," "Visualization," and "User." Data (D) is transformed by a specification (S) into a visualization (V). The Image (*I*) is processed by the perception and cognition of the user (P) to produce knowledge (K), iterated via a time-variant perceptual/cognitive loop (dK/dt), in concert with interactive exploration (IE). Time-variant specification changes (dS/dt) in turn affect V. More explicitly, D undergoes some pre-processing and transformation into an interrogable structure before it is visualized. S includes interactive steps including filtering, mapping, and rendering, affecting the appearance of the visualization. This is pertinent to mapping variables to color, given the interplay between machine models of color spaces (S) and perceptual faculties of users (P) in their context of observation. P implicitly incorporates the context in which a visualization is observed. This includes factors such as ambient illumination conditions and changes in lighting (e.g., shadows and light in a room or daylight through a window). These are all normal criteria for screen and print reproduction quality control in professional digital publishing, and merit greater consideration for well-posed scientific visualization.

Colormaps and Color Scales in Scientific Visualization

Colormaps and color-scales are standard features in interactive scientific visualization software aiming to convey a wide variety of information types: e.g., continuous values, categories and many others (Rheingans, 1992, 2000; Munzner, 2014; Mittelstädt and Keim, 2015; Zhou and Hansen, 2016). Colorization of data can be driven parametrically, e.g., using algorithmic functions (Eisemann et al., 2011), by human aesthetic decisions (Healey and Enns, 2012) or by pre-existing convention and experience (Bertin, 1983; MacEachren et al., 2012). The widely-used rainbow ("*Jet*") or spectrum-approximation colormap, whilst having specific productive use-cases, is well-known for introducing problems of perceptual non-linearity, hue-ordering ambiguity and loss of visual discrimination for fine detail (Rogowitz and Treish, 1998; Eddins, 2014; Hawkins, 2015; Stauffer et al., 2015). Research into optimal colormap design for science has an extensive literature (Silva et al., 2011; Kovesi, 2015; Moreland, 2016; Ware et al., 2018), including optimization for color vision deficiencies (Light and Bartlein, 2004). Addressing the need for consistent terminology, Bujack et al. (2018) propose a nomenclature with unambiguous mathematical definitions, characteristics that are quantifiable via their on-line tool (Bujack et al., 2018).

Colormaps that maximize color difference such as *Viridis*, *Magma*, *Parula* and others are now becoming default schemes in widely-used scientific software (Smith et al., 2015). Ware et al. (2018) and Kovesi (2015) provide thorough analysis of a range of colormaps for different visualization tasks. Crameri (2018a,b) provides extensive discussion of colormaps for geoscientific visualization, as well as software and colormap resources for widely used programs such as GMT, Matlab, QGIS and others.

Human Color Perception

Human color vision is a complex, adaptive system extensively studied by vision researchers (Wyszecki and Stiles, 1982; Gordon, 2004; Stockman and Brainard, 2015). Estimates of the number of discernible colors perceivable by an average human vary widely (Masaoka et al., 2013). The non-uniform nature of human color perception has been well-established by techniques in advanced colorimetry (Fairchild, 2013).

RGB, HSL, HSV Color

A simple computable model for linear RGB color calculates and stores color values as 8-bit values of the three primaries Red, Green, and Blue (RGB), following an additive color-mixing model capable of generating $(2^8)^3$ colors: 16.7 million colors or "24-bit color" (Poynton, 2012). Whilst computationally simple, RGB color mixing is regarded as non-intuitive for end-users (Meier et al., 2004; Zeileis and Hornik, 2006). Tractable transformations of RGB, such as HSL and HSV (Smith, 1978), are simple geometric reformulations of this schematic color model, rather than perception-based ones (Robertson, 1988), and inadequately represent human color perception (Poynton, 2006; Fairchild, 2013). Despite their ease-of-use and ubiquity in computer interfaces, they are discontinuous and not perceptually uniform. They also present problems in accurately representing color and lightness relationships (Light and Bartlein, 2004; Silva et al., 2011; Kovesi, 2015;

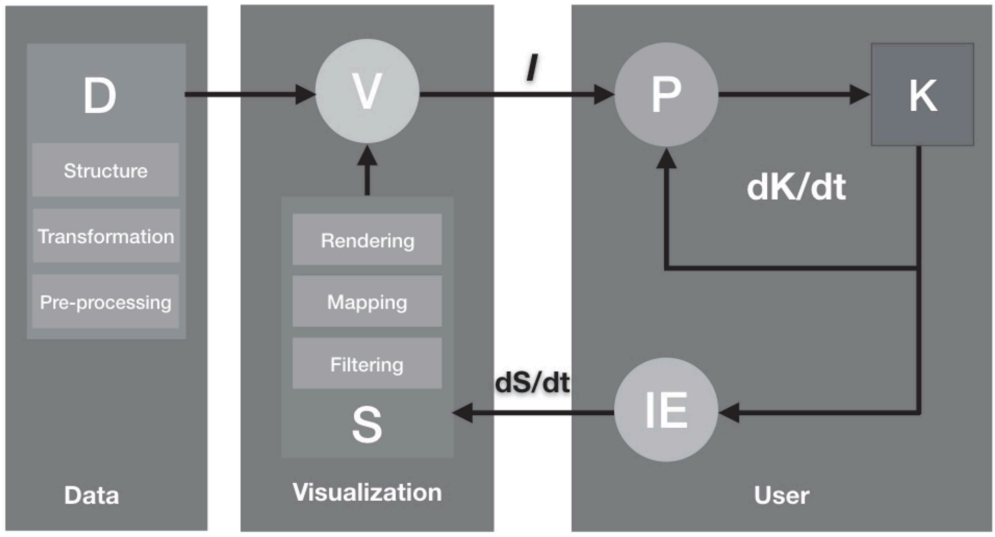


FIGURE 1 | Model of Visualization (after Van Wijk, 2005 and Liu et al., 2014). This figure shows a schematic model of the visualization process, incorporating human-computer interaction.

Moreland, 2016; Ware et al., 2018). Most importantly for data visualization, there is no meaningful metric for representing color difference in RGB, HSL, or HSV color spaces that match human perception of color differences (Robertson, 1988).

CIE Color

The CIE system (CIE, 2004), an international standard for color specification and communication, provides a series of color models that mathematically represent human color perception and color appearance. The CIEXYZ 1931 model is a perceptually measured color space with known values (CIE, 2004). This model represents all colors that are perceivable by an observer with *average* eyesight (Fairchild, 2013; Asano et al., 2016). The range of colors produced by the model is referred to as its *gamut* (Morovic and Luo, 2001). Commonly used color models such as RGB, HSV, HSL, and CMYK exhibit limited gamuts, producing a substantially smaller range of colors than humans are capable of perceiving. Because these models are relative, they require a photometrically defined reference white point, CIE D_{50}/D_{65} (Poynton, 2006), in order for color values to be mapped from one space to another (Poynton, 1994). The standard RGB (sRGB) color space and gamma curve, using CIE defined chromaticities and D_{65} whitepoint, has become the default color space for computer OS and internet color management systems (Anderson et al., 1996; IEC, 1999; Hoffmann, 2000). However, whilst "absolute," sRGB color space is not a perceptual color space and has a significantly smaller gamut than that of CIEXYZ derivatives (Hoffmann, 2000, 2008). Mathematical regularizations of CIEXYZ have led to color models such as CIELAB and others, which closely approximate human color perception (Fairchild, 2013).

Uniform Color Space: CIELAB

Plotting CIE XYZ tristimulus values in Cartesian coordinates produce perceptually non-uniform color spaces (CIE, 2007). Uniform Color Spaces (UCS) are mathematical transformations of the CIE 1931 XYZ gamut that represent color in a perceptually even fashion, defined by the CIE as a color space in which equal metric distances approximately predict and represent equal perceived color differences (Luo et al., 2006; Bujack et al., 2018). As perception-based color models, they more accurately map the human visual gamut and mitigate color-matching and color-difference problems.

CIELAB and other UCS are commonly proposed for creating perceptually uniform color sequences in data visualization (Meyer and Greenberg, 1980; Kovesi, 2015; Ware et al., 2018), despite known limitations and deficiencies (Wyszecki and Stiles, 1982; Sharma and Rodriguez-Pardo, 2012; Fairchild, 2013; Zeyen et al., 2018). They form a set of absolute color spaces defined against reference whites or standard illuminants defined by the CIE (Tkalcic and Tasic, 2003; Foster, 2008; Fairchild, 2013). Due to relative ease of computability, CIELAB has become widely used for color specification and color difference measurement. It provides a complete numerical descriptor of color in a perceptually uniform rectangular coordinate system (Hunter Associates Laboratory Inc., 2018).

Color Difference

The CIELAB UCS color difference metric, ΔE, is calculated as follows (Lindbloom, 2017):

$$E_{ab}{}^{*} = \sqrt{(L_1 - L_2)^2 + (a_1 - a_2)^2 + (b_1 - b_2)^2}$$

for $0 \leq L \leq 100$, $-128 \leq a \leq 127$, $-128 \leq b \leq 127$ (signed 8-bit integer), where L = lightness, a = green (−a) to red (+a), b = blue (−b) to yellow (+b).

Color differences for RGB values are calculated via transposition into CIELAB coordinates within the sRGB gamut, following standard conversion formulae and standard illuminant values (RGB to XYZ, XYZ to CIELAB; Brainard, 2003; Lindbloom, 2013).This provides a dimensionless Euclidean metric for color difference that can be linearly applied to known data values and ranges during color-mapping. ΔE values of ∼2.3 correspond to a just-noticeable-difference (JND) in color stimuli for an average untrained observer (Sharma and Trussell, 1997; Mokrzycki and Tatol, 2011), indicated in **Table 1**. Color opponency can be verified in CIELCh space, using Chroma (C) and Hue (h) values, calculated:

$$C = \sqrt{a^2 + b^2}$$
$$h_{ab} = atan2(b, a)$$

where atan2 is the 2-argument arctangent function.

3D Representations of Color

Most existing computer-based color-palette tools date back to paint programs from the 1980s (Meier et al., 2004). Standard 2D RGB/HSL/HSV color-selection interfaces do not clearly articulate the non-uniformity of human color perception and provide poorly defined feedback on color difference (Douglas and Kirkpatrick, 1999; Stauffer et al., 2015). 1D, 2D, and 3D representations of different color gamuts form an essential part of a user interface for color selection and application (Robertson, 1988; Zeileis and Hornik, 2006). Color gradients can be visualized as paths through representations of two or three-dimensional color spaces. Dimensionality is an imperative consideration in determining the type of path traversal that can be undertaken in a color space: 1D representations implicitly provide no path information, 2D representations address only co-planar colors, 3D representations provide maximal information about path extent, geometry and color relationships (Rheingans and Tebbs, 1990; Bergman et al., 1995). Path traversal is an important indicator for the location of perceptually isoluminant colors, indication of monotonicity (linear increase/decrease in chroma or lightness), quantization or stepping, orthogonality and other salient features (Ware, 1988; Bergman et al., 1995; Rogowitz and Goodman, 2012).

TABLE 1 | ΔE perceptual characteristics (after Mokrzycki and Tatol, 2011).

ΔE	Perceptual Characteristics
0 < ΔE < 1	Observer does not notice the difference
1 < ΔE < 2	Only experienced observer can notice the difference
2 < ΔE < 3.5	Inexperienced observer also notices the difference
3.5 < ΔE < 5	Clear difference in color is noticed
5 < ΔE	Observer notices two different colors

DATA AND METHODS

Interactive Color-Mapping for Geoscience

Interactive color-mapping in an intuitive real-time, performant software application is an appealing proposition for geoscientific data visualization. Interactivity affords immediate visual feedback to the end-user, providing the opportunity to iterate through color palettes and associated colorization functions, exploring available color-spaces and their utility in eliciting features of underlying data. However, great care must be taken to ensure contiguity between data, color, and color-space geometries, including gradient path trajectories.

Although there are many applications that enable the construction of color gradients, few enable live interactive exploration of color spaces whilst being applied to data, concurrently providing visual feedback displaying the gradient path through color space. In this contribution we introduce Gradient Designer (GD), its companion applications and sample colormaps. This suite of tools is suitable not only for color-mapping, gradient design and data exploration, but extend live, real-time interactive visualization beyond the computer desktop to a range of visualization platforms, such as MR, VR, and Dome display systems (Milgram and Kishino, 1994; Morse and Bourke, 2012).

Implementation: Gradient Designer

Gradient Designer (GD) is an interactive gradient design and color-mapping software application aimed at the well-posed display of continuous spatial data, as frequently used in geoscience research. It is implemented on the MacOS platform (Morse, 2019).

GD features the following capabilities:

Data Handling:

- Local or remote datasets maybe be rapidly explored through an interactive interface that allows sequential overview of multiple layers through a 3D dataset, as well as zooming and panning to features of interest.
- Robust and clear relationship to known incoming data values and ranges enabled by UI.
- Export of high-resolution colorized images, as well as color gradients for import into other software in raster image format (.png), color palette table format (.cpt) and data interchange format (JSON).

Color Control:

- RGB gradients can be replicated and analyzed in CIELAB color space.
- Live color space visualization of gradient path traversal in CIELAB, RGB, HSL and HCL color spaces through four companion apps.
- Manipulation of linear RBG/sRGB (D_{65}) gamut colors in 3D CIELAB/RGB/HSL/HCL color spaces using simple HSL slider UI.
- Complex gradients may be designed that target specific values and ranges, including continuous-linear, stepped-linear, non-continuous and non-contiguous ranges.

Extended Functionality:

- Alpha channel control for downstream 3D compositing.
- Live video sharing of color gradient data to Syphon-compatible client applications for display (e.g., on immersive visualization systems).

Application Aims and Development Framework

Gradient Designer (GD) and its companion apps aim to provide an intuitive interface for a set of linear color-mapping tasks for continuous geoscience data. For our case study, data has been pre-processed into whole of globe equirectangular greyscale images stored as 8-bit RGB PNG files, with known data ranges (where 0–255 represent known minima and maxima, linearly mapped to the underlying data, including an alpha channel for lat/long region-of-interest delineation). This provides 256 greyscale values, which are adequate for the data under consideration.

The current version is programmed in the MacOS Quartz Composer VPL, using a mixture of pre-defined QC processing nodes as well as custom routines programmed in Objective-C, Javascript, OpenCL and OpenGL. It is compatible with MacOS 10.13.6 High Sierra and MacOS 10.14 Mojave (Morse, 2019). GD can share live video of the gradient display (**Figure 2**, panel 10) in real-time to external applications running a Syphon-compatible client (Butterworth et al., 2018; NewTek, 2019).

Gradient Designer User Interface

The user interface comprises two windows and standard MacOS menus, providing access to further color picker interfaces at OS-level. The application interface is designed around simple and familiar slider and button controls, and text field inputs. It uses a conventional mouse/keyboard combination.

Figure 2 illustrates the 10 main panels of the application: Panel 1—user instructions; Panel 2, a floating parameters window, provides access to file IO, global colorisation and composite operators, OS-level color pickers, output parameters and D_{65} whitepoint XYZ reference values (IRO Group Ltd, 2019); Panel 3—interactive 3D Viewport; Panel 4—color picker controls—toggle on/off LAB Color Mixer input, GD or OS color picker controls; Panel 5—source image control and metadata; Panel 6—gradient display and design interface including data metric display; Panel 7—gradient control sliders for gradient layers; Panel 8—gradient and colorized data output previews; Panel 9—layer controls; Panel 10—output controls (Morse, 2019). Outputs include the ability to write out gradients as raster image formats (.png), color palette tables (.cpt; Wessel et al., 2013) and data interchange format (JSON), suitable for evaluation via colormeasures.org (Bujack et al., 2018).

Companion Applications

LAB Color Mixer (**Figure 3**) is a companion color-selection app for designing color gradients in CIELAB color space. LAB Color mixer displays RGB and CIELAB gradients in the same view, demonstrating the disparity between interpolation pathways in their respective color spaces. It displays CIELAB ΔE, as well as CIELCh Hue and Chroma values, for verification and color opponency. Companion visualization apps (**Figure 3**) GV_LAB, GV_RGB, GV_HSL, and GV_HCL run a continuous real-time image pixel evaluation, mapping incoming gradient color values to geometric positions in the visualized color spaces. The path through color space is drawn via OpenGL line segments in a looping refresh mode, providing visual feedback on the relationships between gradient termini, hinge-points and vectors in each color space. Dots along the path indicate the number of steps of the incoming gradient. This can be a compute-intensive process, so the detail density of the visualization can be reduced to speed up draw times, depending upon available GPU/CPU resources.

Example: CIELAB Divergent Gradient

Standard 2D color-picker GUIs impart limited information about the relationships between colors in a color space, requiring the user to infer characteristics that would be useful for scientific visualization, such as ΔE. LAB Color Mixer and GV_LAB apps address this gap.

LAB Color Mixer CIELAB gradients quantize in a binary fashion, enabling step ranges between 2 and 128. End termini are mapped first and interpolate toward the central value. For divergent gradients this ensures that gradient steps fall unambiguously either side of the central value. As quantization increases, we asymptotically approach the center value to the point of indistinguishability ($\Delta E < 1$), creating the appearance of a continuous gradient (JND < 1). Colors in a three-point divergent gradient can be selected that maximize color difference, indicated by LAB and ΔE values. ΔE values displayed are rounded to the nearest integer. Unit-level quantization is sufficient for discriminability (see **Table 1**). The UI assists users in defining terminal colors that do not exceed the sRGB gamut by providing gamut warnings (where individual R, G, B values equal or exceed 0 or 255). LAB Color Mixer transmits the CIELAB-conformed color gradients as linear RGB image data via an addressable Syphon server to external client applications.

GV_LAB detects the Syphon server and draws the incoming linear RGB image data in CIELAB space, spatially transposed to the sRGB gamut representation (default: D_{65}, 2° Observer model). GV_LAB visualizes CIELAB color space three-dimensionally, displaying RGB gamut isoluminant colors on the AB plane, L on the vertical axis. This view can be rotated, zoomed and inspected. Path traversal lines between non-adjacent termini indicate where interpolated points may exceed the sRGB gamut. This provides instructive feedback for (a)symmetric gradient design, isoluminance and ΔE interpolation, enabling rapid identification of "problem" gradient regions as users explore the design space.

Case Study: Well-Posed Visualization of Seismic Tomography Depth Slices

As a test case for the visualization of a spatially variable 3D dataset in geoscience, we make use of a published seismic wavespeed model of the Earth's mantle beneath Australia, AuSREM (Kennett et al., 2013). In the following case study, we aim to display a 2D slice through the model in such a way as to:

1: User Instructions.
2: Parameters Window - Data range values, composite operators, OS color pickers, data IO directories, whitepoint XYZ reference values.
3: 3D Viewport - Main display area. Comparison and composite views of colorized and non-colorized data.
4: Color Picker Controls - Reset view and User/System/LAB color picker palette toggle and controls.
Export Image Previews - including Layer selection controls. Image, Color Palette Table and JSON save.
5: Source Image Control - Source file import adjustments - gamma control, linear to sRGB toggle, current active displayed file path, source file colourspace metadata.
6: Gradient Display - working gradient, active gradient region.
7: Gradient Controls - Number of gradient iterations (steps) and quantisation/smoothness; Hue, Saturation, Luminosity, Alpha controls; Toggle on/off gradient layers.
8: Output Previews - image gradient, colorised output, composite product.
9: Layer Controls - image layer separation, active displayed layer.
10: Output Controls - file output (.png, .cpt, .json). Live gradient transmission via Syphon.

FIGURE 2 | Gradient Designer User Interface. The 10 interface panels are listed in the figure and described in further detail in the text.

- minimize the introduction of features that are visually salient, but not relevant to the interpretation.
- reveal distinctive features of the wavespeed in Earth's mantle. This is the most important intent of the visualization.
- manage, in a pragmatic way, the uncertainty in the numerical values.
- explore regions of interest in greater detail.

The Mantle Component of the Ausrem Seismic Tomography Model

The AuSREM model is a mature research product, aimed at capturing the distinctive features of the Earth's mantle for this continental area. It was constructed from several sources, primarily seismic surface wave tomography, supplemented by seismic body wave arrivals and regional tomography. The authors have minimized any artifacts of individual modeling procedures by combining 3D information from multiple sources. AuSREM is therefore a sensible choice of spatially variable dataset to use in the exploration of well-posed visualization approaches.

Data are supplied in the form of numerical seismic wavespeed values in 11 layers from 50 to 300 km, at 25 km intervals. Each layer is gridded at 0.5° intervals between −0.5 and −49.5° latitude, and 105.5 and 179.5° longitude. Wavespeed values are in the range 4.0–4.8 kms^{-1} (Stål, 2019).

Uncertainty

For the purposes of this study, we assume the uncertainty in wavespeed to be constant throughout the model at $\pm$ 0.05 kms^{-1}, i.e., a given value of 4.20 kms^{-1} could be between 4.15 and 4.25 but would not be as small as 4.14 nor as large as 4.26 kms^{-1}. We do not consider spatial uncertainty in this study, brought about by effects such as smearing, noted by Rawlinson et al. (2006). From a pragmatic perspective therefore, for a value at a given point, the uncertainty is the maximum departure from the given value within which an experienced analyst would expect the actual value to be. In the case studies that follow, the contour step interval and other color mapping choices may be set to take account of uncertainty.

Visualizations

GD is used for the case study to create a series of visualizations of the AuSREM dataset, focusing on the 100 km depth slice, which is likely to be representative of the main features of the continental lithosphere. We build upon an example appearing

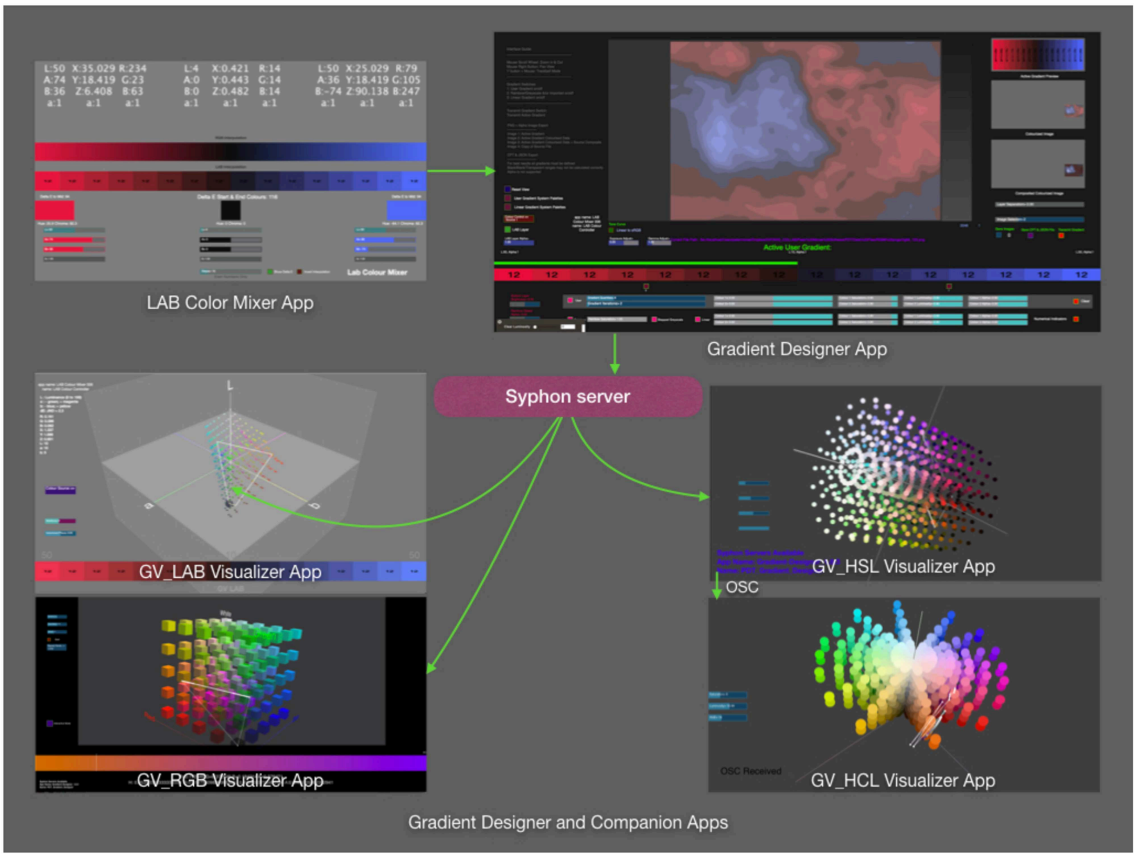


FIGURE 3 | GD Color Space Visualization Companion Apps. This figure illustrates the workflow relationship between the LAB Color Mixer App, Gradient Designer and the four color space visualization apps, GV_LAB, GV_RGB, GV_HSL, and GV_HCL.

in Kennett et al. (2013) (subsequently referred to as KFFY13) and conduct a comparative analysis of our new visualizations. KFFY13 visualizes Earth model reference values at 100 km depth, with wavespeed ranges of 4.00–5.02 kms^{-1}, quantized in 17 steps, each corresponding to a range of 0.06 kms^{-1}. We visualize a wavespeed range of 3.8–5.0 kms^{-1} quantized in 12, 16, 24, 48, 64 steps (0.1, 0.075, 0.05, 0.025, 0.01875 kms^{-1} respectively) as required.

RESULTS

High resolution versions of **Figures 4–8** are available at the link provided in the figure captions. The first visualization (**Figure 4A**) provides a reference view of the model, intended to show the imported model values with a mapping of the underlying values to a continuous gradient, linearly interpolated in RGB color space (transposed to sRGB for display). Subsequent images (**Figures 4B–D**) use a three-point divergent color-map which enables the researcher to examine areas with both low and high values as distinctive features. In **Figure 4B** we replicate the color-mapping used by the AuSREM authors (KFFY13), which is, in many ways a successful visualization. It takes a value close to 4.495 kms^{-1} as its central value, which corresponds to an Earth model reference value at 100 km (Kennett et al., 1995). Two subsequent visualizations (**Figures 4C,D**) make use of companion apps to Gradient Designer, the LAB Color Mixer and GV_LAB, which enable fine-grained control of variation in lightness and color difference (ΔE) within the sRGB gamut. These directly affect how the researcher will perceive features in the image. Numerical feedback provided by the companion apps on LAB values, ΔE and path traversal visualizations assist analysis and reproduction for both 2D and 3D display.

The LAB color mixer directly displays the color difference values between each gradient terminus and the midpoint (ΔE-1, a value pair), the color difference between the two termini (ΔE-2) and ΔE across each interpolated color step. GV LAB visualizes isoluminance, color-map trajectories, ΔE and JNDs in CIELAB color space, clearly indicating the constraints of the sRGB (D_{65}) gamut. These are novel additions to the tools available to the researcher. The divergent sRGB color-map (**Figure 4B**) is replicated (**Figure 4C**) in CIELAB color space, with interval color values interpolated in that space using clearly enumerated ΔE

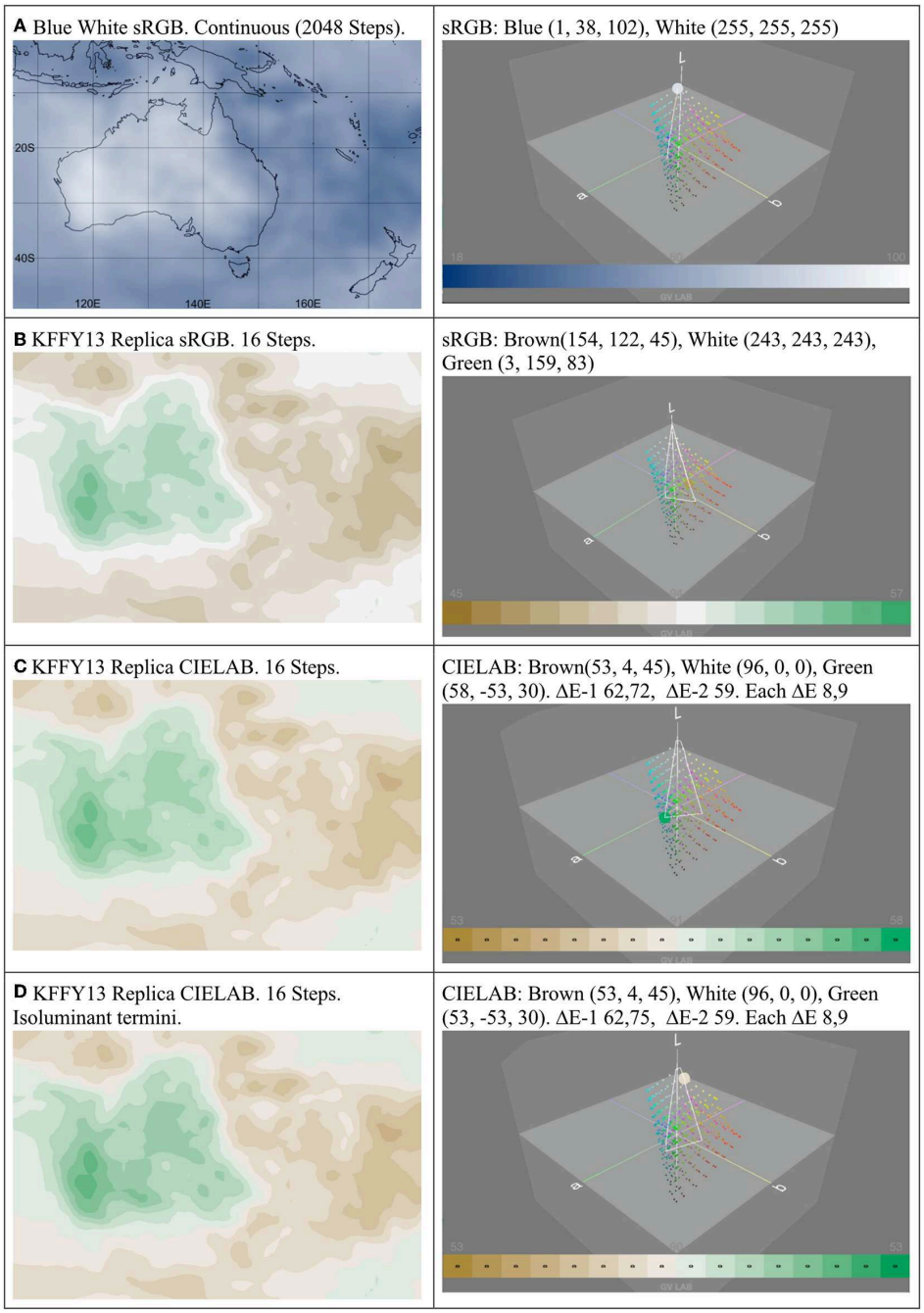


FIGURE 4 | Comparative visualizations generated using Gradient Designer and LAB Color Mixer Apps (left column) shown with supporting insights in CIELAB color space provided through the GV_LAB App (right column). The grid values being displayed are taken from the 100 km depth slice (KFFY13). **(A)** Linearly interpolated HSL/sRGB reference view of the model in near-monochrome with slow values in dark blue; **(B)** Divergent 16 step RGB color-map replicating that used by KFFY13; **(C)** Divergent 16 step CIELAB interpolation replicating **(B)**; **(D)** Divergent 16 step CIELAB interpolation with isoluminant end termini. All images assigned Generic RGB ICC profile. CIELAB colorspace conversions use D_{65} 2° illuminant. High resolution: https://doi.org/10.5281/zenodo.3264037.

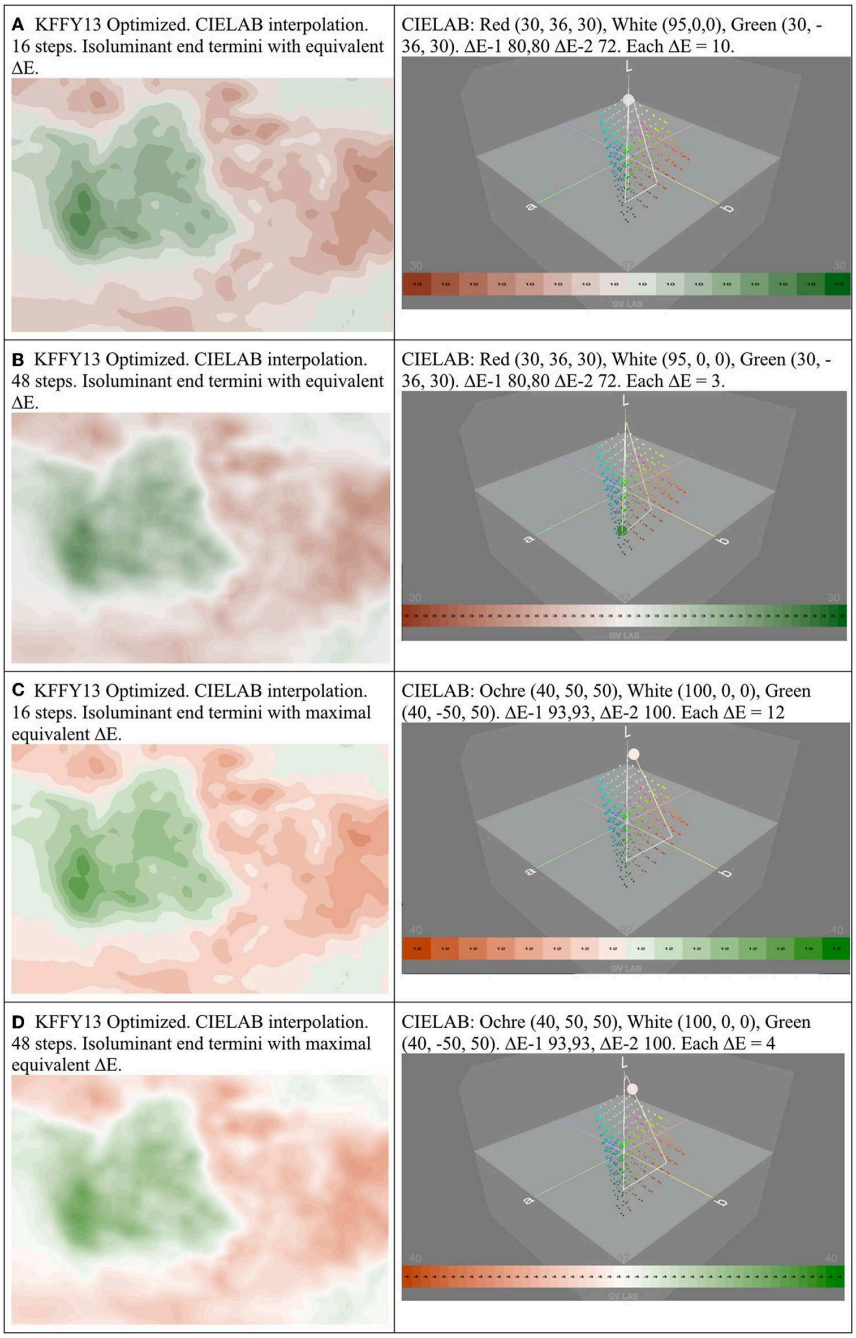


FIGURE 5 | Comparative optimized visualizations generated using the Gradient Designer and LAB Color Mixer Apps (left column) shown with supporting insights in CIELAB color space provided through the GV_LAB App (right column). For details of the grid values being displayed, see the caption for **Figure 4**. **(A)** KFFY13 optimized, 16 step divergent color-map, with isoluminant termini with equivalent ΔE from termini to midpoint; **(B)** As per B, with 48 steps; **(C)** 16 step divergent color-map, with isoluminant termini with enlarged ΔE from termini to midpoint, enlarged ΔE between termini; **(D)** As per **(C)**, with 48 steps. CIELAB colorspace conversions use D_{65} 2° illuminant. Red-Green termini may present difficulties for viewers with deuteranopia, tritanopia. High resolution: https://doi.org/10.5281/zenodo.3264037.

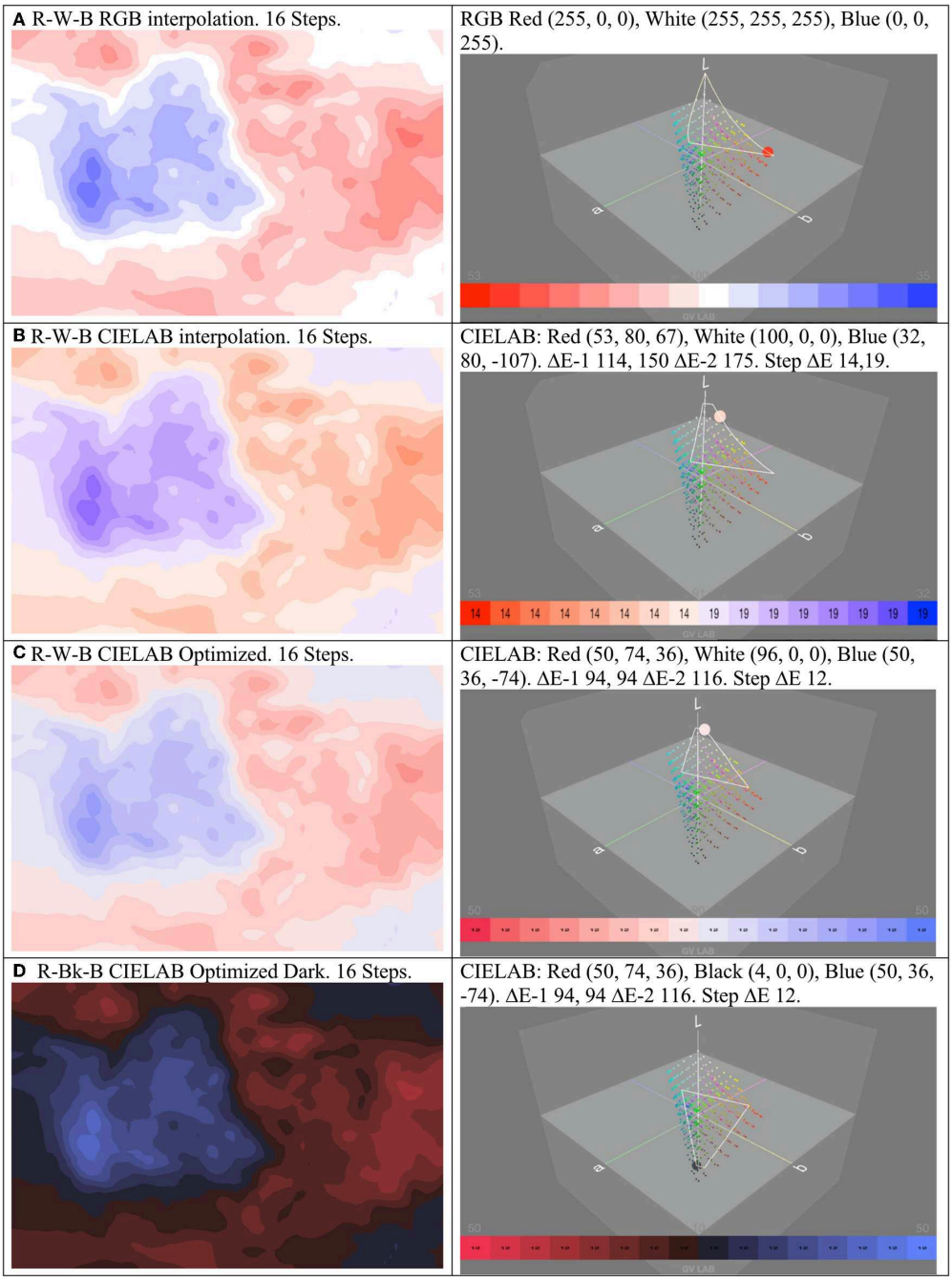


FIGURE 6 | Further comparative visualizations generated using the Gradient Designer and LAB Color Mixer Apps (left column) shown with supporting insights in CIELAB color space provided through the GV_LAB App (right column). **(A)** R-W-B non-linear RGB interpolation 16 Steps; **(B)** R-W-B CIELAB interpolation 16 Steps. Red and Blue Values match 6 **(A)**, Blue-White interpolation is sRGB gamut constrained; **(C)** R-W-B linear CIELAB Optimized 16 Steps, Red and Blue are Isoluminant (L = 50), all colors within sRGB gamut; **(D)** R-Bk-B CIELAB Optimized Dark 16 Steps. Red/Blue as per 6**(C)**, Black L = 4. CIELAB colorspace conversions use D_{65} 2° illuminant. High resolution: https://doi.org/10.5281/zenodo.3264037.

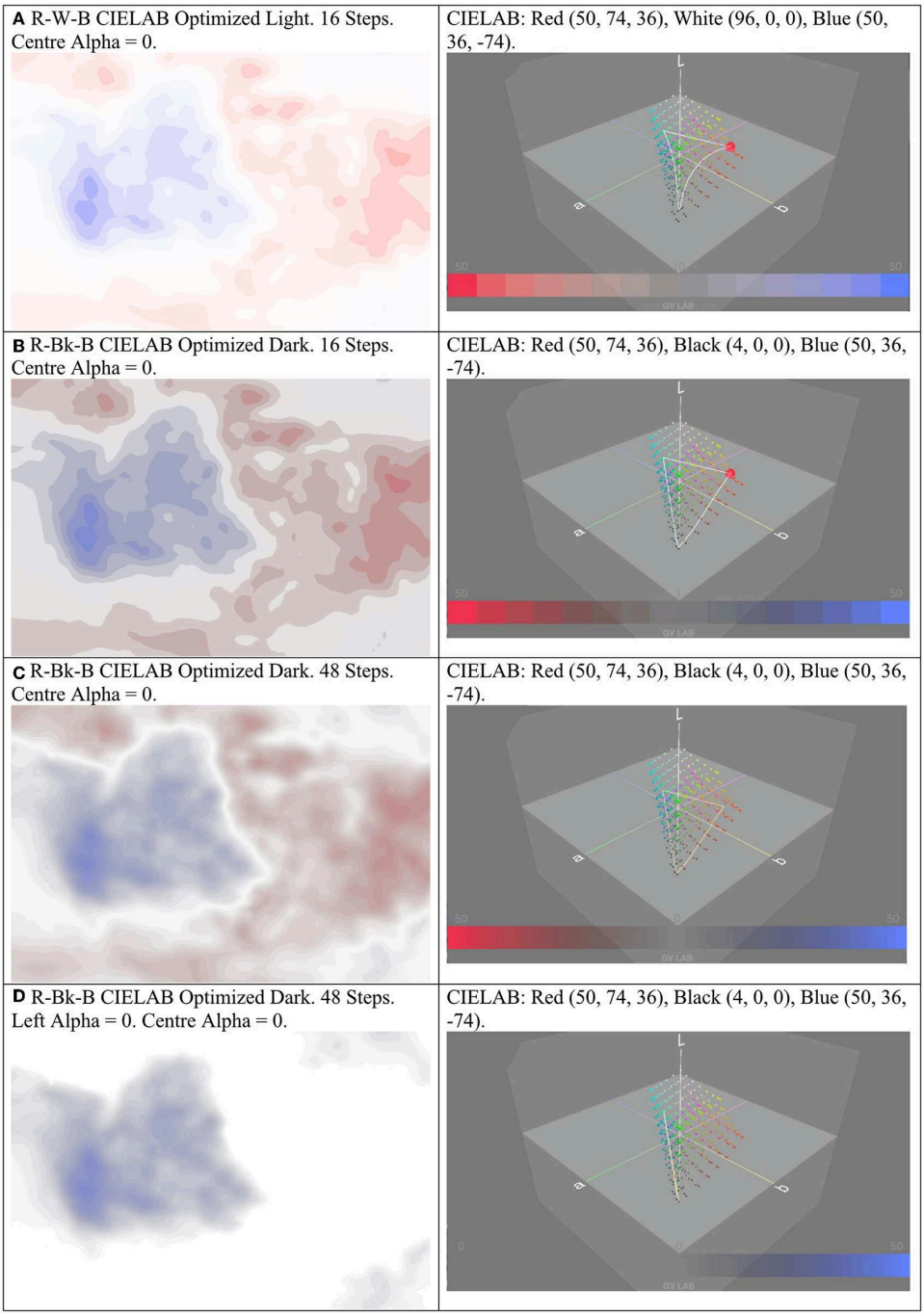


FIGURE 7 | Detailed 2D Views and preparation for 3D. Visualizations generated using the Gradient Designer and LAB Color Mixer Apps (left column) shown with supporting insights in CIELAB color space provided through the GV_LAB App (right column). Note: images contain alpha channels composited against white background. **(A)** R-W-B gradient as per 6 **(C)**, Center Alpha = 0; **(B)** R-Bk-B CIELAB Optimized Dark, 16 steps, Center Alpha = 0; **(C)** R-Bk-B CIELAB Optimized Dark, 48 steps, Center Alpha = 0; **(D)** R-Bk-B CIELAB Optimized Dark. 48 Steps. Left Alpha = 0. Center Alpha = 0. CIELAB colorspace conversions use D_{65} 2° illuminant. High resolution:https://doi.org/10.5281/zenodo.3264037.

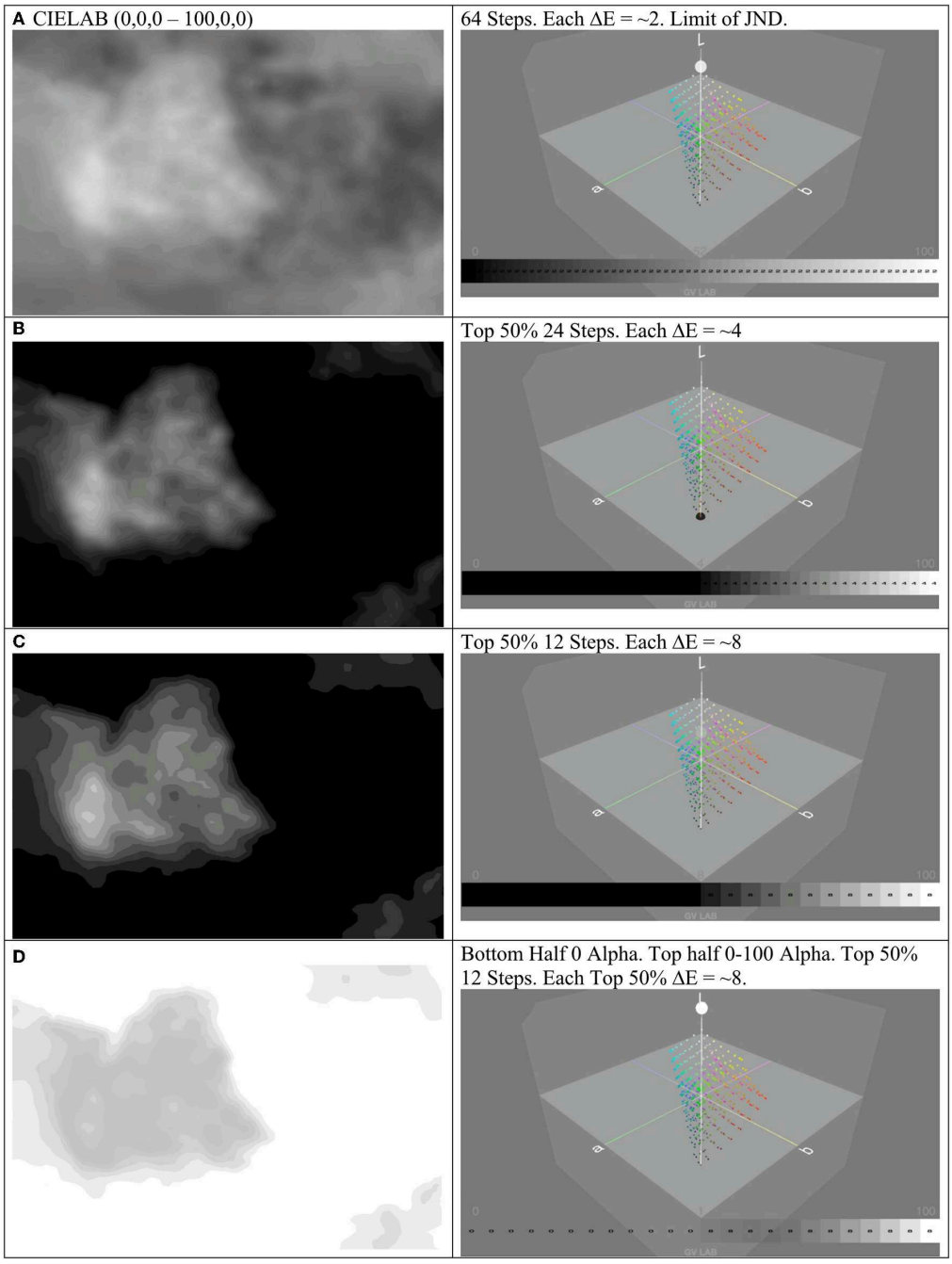


FIGURE 8 | Comparative visualizations, intended to highlight areas of detail, generated using the Gradient Designer App (left column) shown with supporting insights in CIELAB color space provided through the GV_LAB App (right column). For details of the grid values being displayed, see the caption for **Figure 4**. **(A)** Detail All CIELAB 0-90 L 64 Steps. **(B)** Detail top-half CIELAB 0-90L 24 steps. **(C)** Detail top-half CIELAB 0-90L 12 steps. **(D)** Detail top-half CIELAB 0-90L 12 steps + bottom-half alpha 0, top-half alpha 0-100. CIELAB colorspace conversions use D_{65} 2° illuminant. High resolution: https://doi.org/10.5281/zenodo.3264037.

values. In visualization **Figure 4D**, we equalize gradient termini located on an isoluminant AB plane (L = 53). This however, does not guarantee that the color intervals in the divergent three-point gradient cover equivalent Euclidean distance, nor that interpolated colors do not exceed the sRGB gamut.

In **Figures 5A,B**, we optimize the color-map, whilst maintaining the same general color range as **Figure 4D**, ensuring that all interpolated colors fall within the sRGB (D_{65}) gamut. Gradient termini are isoluminant (L = 30) and have equivalized ΔE-1 (=80) for gradient termini. ΔE-2 (=72) is also larger than that exhibited in **Figure 4D** (=59). In **Figures 5C,D** we maximize both ΔE-1 (=93) and ΔE-2 (=100) for that region of the sRGB gamut (visualized by the isosceles triangles on the right column). This increases the dynamic range of the visualization both in lightness (L) and color difference (ΔE), and assures that Euclidean distances for gradient interpolation steps are equivalent, linearly matching the underlying data values. **Figures 5C,D** display results using 16 and 48 steps, with interpolated ΔE values of 12 and 4 respectively. These exceed recommended JND (>3) values for clearly discriminable colors, measurably indicating on which side (- or +) of the reference value they fall. They proceed in linear, quantifiable, perceptually accurate steps mapped to the underlying data values according to the CIELAB color model. At lower (16 steps) quantization the visualization performs categorically, where color steps implicitly consolidate wavespeed ranges and uncertainty. In concert with a higher dynamic range in L and ΔE, increased (finer) quantization acts like a lens, sharpening focus upon the underlying data, eliciting physical structural information to the extent this can be visually inferred within the limits of known uncertainty.

In **Figures 6A–D** Red-White-Blue divergent gradient colors are chosen to align with convention in seismology (i.e., reddish/orange colors represent slow wavespeeds; bluish colors represent fast wavespeeds). Terminal color values and their equivalents in different color spaces appear in **Table 2**. Following Kovesi (2015, p. 17), **Figure 6A** presents a standard RGB Red-White-Blue divergent gradient interpolated in RGB color space. Visualizing this in GV_LAB (right-hand column) in CIELAB color space clearly reveals this to be asymmetrical, non-linearly interpolated and volumetrically constrained by the sRGB gamut. **Figure 6B** replicates 6A in CIELAB colorspace, with Red/Blue terminal values per **Table 2**. Steps between termini are linearly interpolated within the sRGB volume, with Step ΔE-2 = 14,19. End termini to midpoint quantization constrains the point of closest interpolation to white. The gradient is non-optimal due to non-isoluminant end termini, sRGB volumetric constraint and asymmetry. None of the values in the underlying data map linearly to the Red/Blue extrema, implying that 6A is a poor representation.

Figure 6C optimizes 6B in CIELAB, adjusting color values to more closely match 6A. Red/Blue values are isoluminant (L = 50), all interpolated colors fall linearly (Step ΔE = 12) within the sRGB volume, forming a perceptually symmetrical gradient (ΔE-1 = 94, 94). The balanced dynamic range, linearly matching color differences and symmetry ensure a quantifiable perceptual match with underlying data values. Setting Red/Blue values to L = 50 also ensures that the gradient is easily invertible within the sRGB gamut.

Figure 6D sets the midpoint reference value to near Black (L = 4), and maintains Red/Blue CIELAB values from 6C. This maintains the ΔE-1, ΔE-2, and Steps ΔE of **Figure 6C**. It is perceptually near-equivalent to **Figure 6C**, with judiciously chosen termini within the complex shape of the sRGB volume. Invertibility of a gradient is a desirable characteristic for downstream 3D compositing methodologies (Porter and Duff, 1984), and should be tested for dynamic range and gamut exceedance at the outset of the design-decision process.

In **Figures 7A–D**, we introduce the ability of the LAB Color Mixer App to assign an interpolated alpha channel data to the gradient. This is not driven by underlying seismic wavespeed values directly, but by the user: the alpha channel is linearly interpolated across the number of gradient steps, and in this case enables specification of transparency. **Figure 7A** illustrates the non-linear interpolation this introduces when an alpha channel is applied to the optimized Red-White-Blue gradient. This is a consequence of the alpha blending function in OpenGL: multiplication of alpha values (0-1) applied to the relevant color values as L increases (Telea, 2007). **Figure 7B** demonstrates the retention of linearity when colors interpolate to black rather than white, affirming that an invertible divergent gradient is desirable for a variety of compositing approaches, dependent upon context and the compositing algorithm chosen. **Figure 7C** demonstrates the effect of finer gradient steps, which may be desirable in tomographic visualization. Finally, **Figure 7D** demonstrates the ability to display a selected range of the data (in this case the upper 50%) using alpha information, with the concomitant GV_LAB plot representing this as a straight line, as expected. These outputs are suitable for 3D compositing of multiple layers in external applications.

Our final visualizations in **Figures 8A–D** dispense with color and apply a simple greyscale gradient, linearly interpolated in CIELAB color space, with the intent to use lightness to reveal shape from shading. In this instance features emerge corresponding to regions of wavespeed contiguity and other structures of the mantle. **Figure 8A** illustrates the entire model with L ranging from 0 to 100, in 64 steps. This reduces the ΔE of each interpolated step to approximately 2, approaching the limit of JND for an expert observer. **Figure 8B** highlights the top 50% of the range, indicating regions of high wavespeed (24 steps), with each step ΔE = 4. A coarser approximation (12 steps) in **Figure 8C** reveals clear groupings within this subset, with a ΔE per step of ∼8. Finally, **Figure 8D** introduces an alpha channel, applied at 100% for the lower 50% of the data, and linearly stepped from 0 to 100% over the top 50% range. This similarly prepares the output layers for 3D display.

DISCUSSION

In the following section we note current limitations in our software and data visualization pipeline, followed by an appraisal of the strength of our approaches and the on-going potential for future research and development.

TABLE 2 | Figures 6A–D Gradient Color Values.

Figures	Color	RGB			HSL (normalized)			CIELAB (D_{65})			CIELCh (D_{65})	
		R	G	B	H	S	L	L	A	B	C	h
Figures 6A,B	Red	255	0	0	0	1	0.5	53	80	67	104.4	39.9°
	White	255	255	255	0	0	1	100	0	0	0	270°
	Blue	0	0	255	0.7	1	0.5	32	79	−107	133.8	306.3°
Figure 6C	Red	234	23	63	1	0.8	0.5	50	74	36	82.3	25.9°
	White	243	243	243	0	0	1	96	0	0	0	0°
	Blue	79	105	247	0.6	0.9	0.6	50	36	−74	82.3	−64.1°
Figure 6D	Red	234	23	63	1	0.8	0.5	50	74	36	82.3	25.9°
	Black	14	14	14	0	0	0.1	4	0	0	0	0°
	Blue	79	105	247	0.6	0.9	0.6	50	36	−74	82.3	−64.1°

RGB 0-255, HSL Normalized 0-1.00, CIELAB and CIELCh D_{65}.

Implementing our software in the QC VPL limits our software to the MacOS platform. QC was chosen for its simple visual programming paradigm, its facility for rapid application prototyping and wide API support. Since 2018 QC has been deprecated by Apple, and future support is unclear. We intend to reimplement our software in an alternative VPL (e.g., *Touch Designer*, Derivative Inc., 2019) or a game engine environment (e.g., Unity Technologies, 2019), which include cross-platform support. Aspects of the UI can be improved subject to user-feedback. Future development could incorporate more sophisticated color models (e.g., CIECAM02; Fairchild, 2013), to account for known deficiencies with CIELAB. The 8-bit pipeline can be extended to 16- or 32-bit color and greyscale, including float as well as integer values for greater precision in the manipulation of continuous data. Improvements in alpha channel control could include evaluation of color-shift due to transparency mapping, extra functions for the alpha channel (e.g., tagging or delimiting regions, defining isosurfaces). Output improvements could include more control over Syphon/OSC output, including addressable transfer functions suitable for true volumetric raycasters and other shading models.

Our visualizations demonstrate that color-mapping should be conducted with great care and that extant mappings can be improved. Building on Kovesi (2015), Ware et al. (2018) and Crameri (2018a), we employed CIELAB color space to visualize continuous geoscience data.

We first showed a naive two color HSL/sRGB gradient reference visualization (**Figure 4A**) for the 100 km depth slice of the AuSREM model. **Figure 4B** closely replicated the sRGB KFFY13 100 km depth slice visualization incorporating uncertainty. Visualizing this gradient within the sRGB gamut mapped within CIELAB color space revealed the non-linear interpolation of the gradient. The gradient termini were re-mapped within CIELAB space, resulting in linear color interpolation between gradient termini. This revealed the difference in lightness (L) values between the end-point colors, in turn demonstrating that color differences between end termini and midpoint were non-equivalent. Using CIELAB color space we can accurately quantify this non-equivalence in ΔE values, both for the end-midpoint ranges and for the color difference of each step in the gradient. This causes the end points of the gradient to have different "perceptual distances" from the reference midpoint value, potentially creating the visual inference that regions of slower wavespeed are more proximate to the reference value than regions of higher wavespeed, when in fact this is not the case. **Figure 4D** adjusts for end-point isoluminance, but does not equalize ΔE, demonstrating that despite matching the L values of the endpoints, because color is used in the visualization, adjustments to color termini in the AB plane must also be undertaken in order to ensure gradient symmetry and matching of ΔE values.

Our software enabled us to control precisely and regularize these CIELAB differences, and to characterize uncertainty, using the ΔE metric. Isoluminant gradient termini, equidistant from the midpoint reference value, were established in **Figure 5A**. **Figure 5B** quantized the gradient from 16 to 48 steps, approaching the non-professional observer JND discriminability limit of $2 \leq \Delta E \leq 3.5$ per step. Whilst this exceeds our nominal uncertainty of $+/-$ 0.05 kms^{-1} per step, it affirms that coarser approximations will encode discriminable uncertainty across the gradient and consistently visually agglomerate regions according to a linear ΔE metric.

Optimizing for L and AB variation to elicit form, **Figures 5C,D** take advantage of the ability of our software to symmetrically maximize ΔE within the sRGB gamut. This demonstrates the capacity to linearly sharpen the chromatic distinction between steps, as well as maximizing lightness variation between end and midpoints of the gradient. This has the effect of enhancing contrast, shape perception and slow-fast discrimination across the data slice, whilst maintaining a JND of between $\Delta E = 12$ and $\Delta E = 4$, values that are well within the capabilities of the non-expert observer.

A standard seismic Red-White-Blue divergent gradient is optimized in **Figures 6A,B**, clearly demonstrating the non-linearity of this gradient if it is interpolated in RGB colorspace or naively transposed to CIELAB. This suggests that visualizations that naively use similar gradients may be misleading. **Figure 6C** represents a controlled, isoluminant, ΔE equivalized CIELAB version, with correct linear interpolation. The attraction of this approach is illustrated in **Figure 6D**, where the gradient is inverted, but is perceptually isometric. Isometric invertibility is

attractive for 3D compositing operations, some 2D display and print operations, as well as for volumetric visualization.

Alpha is an important feature for compositing operations and may have unintended consequences for color perception. **Figure 7** demonstrates the use of linearly interpolated alpha channels and their effect upon color values within CIELAB. **Figures 7B,C** demonstrates the retention of color interpolation linearity by the inverted gradient, which is desirable for additive or multiplicative compositing operations. ΔE values are disregarded when premultiplied alpha is applied. **Figure 7D** illustrates the capability of isolating value ranges through the use of alpha channel control.

The utility of CIELAB greyscale interpolation is demonstrated in **Figure 8**. Unlike perceptually non-uniform RGB greyscale interpolation, CIELAB greyscale is perceptually linear and midpoint gray is accurately represented (L = 50). The capability of LAB color mixer, GV_LAB and Gradient Designer is demonstrated in resolving the AusREM data from the limits of discriminability (**Figure 8A**), to the limits of known uncertainty (**Figure 8B**), along with the ability to isolate regions of interest (**Figures 8C,D**).

Visual inference may be one of the first important steps in ascertaining significant formal aspects of geoscientific data. Looking for structure in noisy or uncertain continuous data requires clarity and precision in analytical techniques, including color-mapping. Our software suite, repeatable metrics, and illustrative color-maps are a proof-of-concept that illuminate the relationships between data variables and color that should be part of visually-aware science and human-computer interaction for visual analytics. Interfaces for colorisation using perceptually uniform color spaces, such as CIELAB, provide greater certainty for the accurate visualization of data as well as enhancing the reproducibility of results.

CONCLUSIONS

We have demonstrated the subtle problems that the inherent non-uniformity of common RGB-based color models may present for the interpretation of scientific visualization. Our approach of color-mapping using the CIELAB color model improves the correspondence between a color-map and underlying data. We have demonstrated the capacity of interactive software to apply linear, quantifiable, and perceptually accurate color to a typical geoscience dataset, finding that:

1) Color should be applied to visualization of data with care.
2) CIELAB is a good choice of color space for colorization of linear data as it closely matches human color perception and facilitates a linear mapping between color space metrics and underlying data. Understanding that the sRGB gamut is a constrained subset of the CIELAB gamut is important knowledge. Visualization activities should take this constraint into account at the outset of the visualization process.
3) The ΔE metric accounts for linear color difference and establishes the limits of discriminability for color differences perceivable by the average observer. If color is to be interpolated across underlying data values, then this should take place within an appropriate linear color model, such as CIELAB, within the sRGB gamut.
4) Uncertainty can be characterized using ΔE. In this way, the visualization captures features of the data as well as implicitly representing the uncertainty, encoded as color difference.
5) Form is best expressed using lightness variation in CIELAB colorspace. Lightness can be linearly mapped to underlying data values, using dynamic range to elucidate spatial features. It may be used together with chromaticity or independently as greyscale. The interplay between chromaticity and lightness must be balanced in an effective visualization and may be explored using the software presented.
6) Reproducibility of color is enabled through the use of CIELAB. All color reproduction devices and media are susceptible to color variance. As a perceptual color space, CIELAB is an absolute color space with known values. Stipulating LAB and ΔE values for applied gradients in scientific visualization achieves the desirable goal of accurate color reproducibility across a range of platforms and media, such as calibrated computer displays and print devices.
7) Our contribution encourages attention to both "visual literacy" and "visual numeracy" for scientific data visualization. In providing software, raster image output files (.png), color palette files (.cpt) and data interchange outputs (JSON) that enable linear, quantifiable and perceptually accurate color, we hope to promote well-posed scientific visualization.

DATA AVAILABILITY STATEMENT

Color-maps in raster image format (.png), color palette table (.cpt), and data interchange format (JSON), Gradient Designer software and companion applications are available for download from: https://doi.org/10.5281/zenodo.3264037.

AUTHOR CONTRIBUTIONS

PM co-designed the study, developed the visualization software and wrote the text. AR co-designed the study and contributed to the text. TS carried out data pre-processing and contributed to the text.

FUNDING

PM acknowledges an RTP Scholarship from the University of Tasmania. TS acknowledges support under Australian Research Council's Special Research Initiative for Antarctic Gateway Partnership (Project ID SR140300001). This research was partly supported by the ARC Research Hub for Transforming the Mining Value Chain (project number IH130200004).

ACKNOWLEDGMENTS

The authors thank Paul Bourke for providing javascript programming expertise, Tom Ostersen for GMT expertise, and Stephen Walters for constructive feedback.

REFERENCES

Anderson, M., Motta, R., Chandrasekar, S., and Stokes, M. (1996). "Proposal for a standard default color space for the internet – sRGB," in *Color and Imaging Conference, Vol. 1996* (Scottsdale, AZ: Society for Imaging Science and Technology), 238–245.

Asano, Y., Fairchild, M. D., and Blondé, L. (2016). Individual colorimetric observer model. *PLoS ONE* 11:e0145671. doi: 10.1371/journal.pone.0145671

Bergman, L. D., Rogowitz, B. E., and Treinish, L. A. (1995). "A rule-based tool for assisting colormap selection," in *Proceedings of the 6th Conference on Visualization'95* (Atlanta, GA: IEEE Computer Society). doi: 10.1109/VISUAL.1995.480803

Bertin, J. (1983). *Semiology of Graphics: Diagrams, Networks, Maps*, *Vol. 1*. Transl. by W. J. Berg. Madison, WI: University of Wisconsin Press.

Brainard, D. H. (2003). "Color appearance and color difference specification," in *The Science of Color,* ed S. K. Shevell (Amsterdam: Elsevier), 191–216. doi: 10.1016/B978-044451251-2/50006-4

Buja, A., Cook, D., Hofmann, H., Lawrence, M., Lee, E.-K., Swayne, D. F., et al. (2009). Statistical inference for exploratory data analysis and model diagnostics. *Philos. Trans. R. Soc. Lond.* 367, 4361–4383. doi: 10.1098/rsta.2009.0120

Bujack, R., Turton, T. L., Samsel, F., Ware, C., Rogers, D. H., and Ahrens, J. (2018). The good, the bad, and the ugly: a theoretical framework for the assessment of continuous colormaps. *IEEE Trans. Visual. Comput. Graph.* 24, 923–933. doi: 10.1109/TVCG.2017.2743978

Butterworth, T., Marini, A., and Contributors, S. (2018). *Syphon Framework [Objective-C, MacOS]*. Available online at: https://github.com/Syphon/syphon-framework (accessed May 28, 2019).

CIE (2004). *Technical Report: Colorimetry*, *3rd Edn*, *Vol. 15*. Commission International de L'Eclairage.

CIE (2007). *Colorimetry — Part 4: CIE 1976 L*a*b* Colour Space*. CIE. Available online at: http://www.cie.co.at/publications/colorimetry-part-4-cie-1976-lab-colour-space (accessed February 5, 2019).

Cook, D. (2017). *Myth Busting and Apophenia in Data Visualisation: Is What You See Really There*? Available online at: http://dicook.org/talk/ihaka/ (accessed February 12, 2019).

Cook, D., Lee, E.-K., and Majumder, M. (2016). Data visualization and statistical graphics in big data analysis. *Annu. Rev. Stat. Appl.* 3, 133–159. doi: 10.1146/annurev-statistics-041715-033420

Crameri, F. (2018a). Geodynamic diagnostics, scientific visualisation and StagLab 3.0. *Geosci. Model Dev.* 11, 2541–2562. doi: 10.5194/gmd-11-2541-2018

Crameri, F. (2018b). *Scientific Colour Maps* (Oslo). doi: 10.5281/zenodo.2649252

Derivative Inc. (2019). *Derivative TouchDesigner*. Available online at: https://www.derivative.ca/ (accessed May 8, 2019).

Douglas, S. A., and Kirkpatrick, A. E. (1999). Model and representation: the effect of visual feedback on human performance in a color picker interface. *ACM Trans. Graph.* 18, 96–127. doi: 10.1145/318009.318011

Eddins, S. (2014). *Rainbow Color Map Critiques: An Overview and Annotated Bibliography*. Available online at: https://au.mathworks.com/company/newsletters/articles/rainbow-color-map-critiques-an-overview-and-annotated-bibliography.html (accessed July 2, 2019).

Eisemann, M., Albuquerque, G., and Magnor, M. (2011). "Data driven color mapping," in *Proceedings of EuroVA: International Workshop on Visual Analytics* (Bergen: Eurographics Association).

Fairchild, M. D. (2013). *Color Appearance Models*. Chichester, UK: John Wiley and Sons. doi: 10.1002/9781118653128

Foster, D. H. (eds.). (2008). "Color appearance," in *The Senses: A Comprehensive Reference, Vol. 2*, eds A. Basbaum, A. Kaneko, G. Shepherd, G. Westheimer, T. Albright, and R. Masland (San Diego, CA: Academic Press), 119–132. doi: 10.1016/B978-012370880-9.00303-0

Gordon, I. E. (2004). *Theories of Visual Perception, 3 Edn*. Hove: Psychology Press. doi: 10.4324/9780203502259

Haber, R. B., and McNabb, D. A. (1990). Visualization idioms: a conceptual model for scientific visualization systems. *Visual. Sci. Comput.* 74:93.

Hansen, C. D., and Johnson, C. R. (eds.). (2005). *The Visualization Handbook*. Boston, MA: Elsevier-Butterworth Heinemann.

Harold, J., Lorenzoni, I., Shipley, T. F., and Coventry, K. R. (2016). Cognitive and psychological science insights to improve climate change data visualization. *Nat. Clim. Change* 6, 1080–1089. doi: 10.1038/nclimate3162

Hawkins, E. (2015). Graphics: scrap rainbow colour scales. *Nature* 519:291. doi: 10.1038/519291d

Healey, C. G., and Enns, J. T. (2012). Attention and visual memory in visualization and computer graphics. *Visual. Computer Graph. IEEE Trans.* 18, 1170–1188. doi: 10.1109/TVCG.2011.127

Hoffmann, G. (2000). *CIE Color Space [Professional]*. Available online at: http://www.docs-hoffmann.de/ciexyz29082000.pdf (accessed July 2, 2019).

Hoffmann, G. (2008). *CIELab Color Space [Professional]*. Available online at: http://www.docs-hoffmann.de/cielab03022003.pdf (accessed July 2, 2019).

Hunter Associates Laboratory Inc. (2018). *Brief Explanation of delta E or delta E**. Available online at: http://support.hunterlab.com/hc/en-us/articles/203023559-Brief-Explanation-of-delta-E-or-delta-E- (accessed February 18, 2019).

IEC (1999). *IEC 61966-2-1:1999. Multimedia Systems and Equipment - Colour Measurement and management - Part 2-1: Colour Management - Default RGB Colour Space - sRGB*. Available online at: https://webstore.iec.ch/publication/6169 (accessed May 3, 2019).

IRO Group Ltd (2019). *Math|EasyRGB*. Available online at: http://www.easyrgb.com/en/math.php (accessed May 23, 2019).

Keim, D. A. (2001). Visual exploration of large data sets. *Commun. ACM* 44, 38–44. doi: 10.1145/381641.381656

Keim, D. A., Kohlhammer, J., Ellis, G., and Mansmann, F. (2010). "Mastering the information age solving problems with visual analytics," in *Eurographics*, eds D. Keim, J. Kohlhammer, G. Ellis, and F. Mansmann. Retrieved from: http://diglib.eg.org

Keim, D. A., Mansmann, F., Schneidewind, J., and Ziegler, H. (2006). "Challenges in visual data analysis," in *Tenth International Conference on Information Visualisation (IV'06)* (London, UK), 9–16. doi: 10.1109/IV.2006.31

Kennett, B. L. N., Engdahl, E. R., and Buland, R. (1995). Constraints on seismic velocities in the Earth from traveltimes. *Geophys. J. Int.* 122, 108–124. doi: 10.1111/j.1365-246X.1995.tb03540.x

Kennett, B. L. N., Fichtner, A., Fishwick, S., and Yoshizawa, K. (2013). Australian Seismological Reference Model (AuSREM): mantle component. *Geophys. J. Int.* 192, 871–887. doi: 10.1093/gji/ggs065

Kovesi, P. (2015). Good colour maps: how to design them. *arxiv:1509.03700 [Cs]*. Available online at: http://arxiv.org/abs/1509.03700 (accessed May 28, 2019).

Light, A., and Bartlein, P. J. (2004). The end of the rainbow? Color schemes for improved data graphics. *Eos Trans. Am. Geophys. Union* 85, 385–391. doi: 10.1029/2004EO400002

Lindbloom, B. (2013). *Useful Color Equations*. Available online at: http://www.brucelindbloom.com/index.html (accessed February 18, 2019).

Lindbloom, B. (2017). *Delta E (CIE 1976)*. Retrieved from: http://www.brucelindbloom.com/index.html?Eqn_DeltaE_CIE76.html (accessed February 18, 2019).

Liu, S., Cui, W., Wu, Y., and Liu, M. (2014). A survey on information visualization: recent advances and challenges. *Vis. Comput.* 30, 1373–1393. doi: 10.1007/s00371-013-0892-3

Luo, M. R., Cui, G., and Li, C. (2006). Uniform colour spaces based on CIECAM02 colour appearance model. *Color Res. Appl.* 31, 320–330. doi: 10.1002/col.20227

MacEachren, A. M., Roth, R. E., O'Brien, J., Li, B., Swingley, D., and Gahegan, M. (2012). Visual semiotics and uncertainty visualization: an empirical study. *IEEE Trans. Visual. Comput. Graph.* 18, 2496–2505. doi: 10.1109/TVCG.2012.279

Masaoka, K., Berns, R. S., Fairchild, M. D., and Moghareh Abed, F. (2013). Number of discernible object colors is a conundrum. *J. Opt. Soc. Am.* 30:264. doi: 10.1364/JOSAA.30.000264

Meier, B. J., Spalter, A. M., and Karelitz, D. B. (2004). Interactive color palette tools. *IEEE Comput. Graph. Appl.* 24, 64–72. doi: 10.1109/MCG.2004.1297012

Meyer, G. W., and Greenberg, D. P. (1980). Perceptual color spaces for computer graphics. *ACM SIGGRAPH Comput. Graph.* 14, 254–261. doi: 10.1145/965105.807502

Milgram, P., and Kishino, F. (1994). "A taxonomy of mixed reality visual displays," in *IEICE Transactions on Information System E77-D(12)* (Tokyo), 16.

Mittelstädt, S., and Keim, D. A. (2015). Efficient contrast effect compensation with personalized perception models. *Comput. Graph. Forum* 34, 211–220. doi: 10.1111/cgf.12633

Mokrzycki, W. S., and Tatol, M. (2011). Colour difference ΔE - A survey. *Mach. Graph. Vis.* 20, 383–411. Retrieved from: https://www.researchgate.

net/publication/236023905_Color_difference_Delta_E_-_A_survey (accessed October 15, 2019).

Moreland, K. (2016). Why we use bad color maps and what you can do about it. *Electron. Imaging* 2016, 1–6. doi: 10.2352/ISSN.2470-1173.2016.16.HVEI-133

Morovic, J., and Luo, M. R. (2001). The fundamentals of gamut mapping: a survey. *J. Imaging Sci. Technol.* 45, 283–290. Retrieved from: https://www.researchgate.net/publication/236121201_The_Fundamentals_of_Gamut_Mapping_A_Survey (accessed October 15, 2019).

Morse, P. E. (2019). *Pemorse/Data-Visualisation-Tools: Gradient Designer Suite. V.0.9-Alpha Release*. Hobart, TAS. doi: 10.5281/zenodo.3264037

Morse, P. E., and Bourke, P. D. (2012). "Hemispherical dome adventures," *Presented at iVEC* (University of Western Australia). Retrieved from: http://rgdoi.net/10.13140/RG.2.2.15942.16963

Munzner, T. (2014). *Visualization Analysis and Design*. Boca Raton, FL: A. K. Peters/CRC Press.

NewTek (2019). *NewTek NDI Tools*. Available online at: https://www.newtek.com/ndi/tools/ (accessed February 22, 2019).

Porter, T., and Duff, T. (1984). Compositing digital images. *Comput. Graph.* 18:7. doi: 10.1145/964965.808606

Poynton, C. A. (1994). "Wide gamut device-independent colour image interchange," in *International Broadcasting Convention - IBC '94* (Amsterdam), 218–222. doi: 10.1049/cp:19940755

Poynton, C. A. (2006). *Frequently asked questions about color [Professional]*. Available online at: http://poynton.ca/ColorFAQ.html (accessed May 28, 2019).

Poynton, C. A. (2012). *Digital Video and HD: Algorithms and Interfaces*. Amsterdam: Elsevier. doi: 10.1016/B978-0-12-391926-7.50063-1

Rawlinson, N., Reading, A. M., and Kennett, B. L. N. (2006). Lithospheric structure of Tasmania from a novel form of teleseismic tomography. *J. Geophys. Res.* 111:B02301. doi: 10.1029/2005JB003803

Rheingans, P. (1992). "Color, change, and control of quantitative data display," in *IEEE Conference on Visualization, 1992, Visualization'92* (Boston, MA: IEEE), 252–259. doi: 10.1109/VISUAL.1992.235201

Rheingans, P. (2000). "Task-based color scale design," in *28th AIPR Workshop: 3D Visualization for Data Exploration and Decision Making*, *Vol. 3905*, 35–44. Retrieved from: http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=916058

Rheingans, P., and Tebbs, B. (1990). A tool for dynamic explorations of color mappings. *ACM SIGGRAPH Comput. Graph.* 24, 145–146. doi: 10.1145/91394.91436

Robertson, P. K. (1988). Visualizing color gamuts: a user interface for the effective use of perceptual color spaces in data displays. *IEEE Comput. Graph. Appl.* 8, 50–64. doi: 10.1109/38.7761

Rogowitz, B. E., and Goodman, A. (2012). "Integrating human- and computer-based approaches for feature extraction and analysis," in *Proceedings of the SPIE* (Burlingame, CA), 8291. doi: 10.1117/12.915887

Rogowitz, B. E., and Treish, L. A. (1998). Data visualisation: the end of the rainbow. *IEEE Spectr.* 35, 52–59. doi: 10.1109/6.736450

Rougier, N. P., Droettboom, M., and Bourne, P. E. (2014). Ten simple rules for better figures. *PLoS Computat. Biol.* 10:e1003833. doi: 10.1371/journal.pcbi.1003833

Schulz, H., Nocke, T., Heitzler, M., and Schumann, H. (2013). A design space of visualization tasks. *IEEE Trans. Visual. Comput. Graph.* 19, 2366–2375. doi: 10.1109/TVCG.2013.120

Sharma, G., and Rodriguez-Pardo, C. E. (2012). "The dark side of CIELAB," in *Color Imaging XVII: Displaying, Processing, Hardcopy, and Applications*, *Vol. 8292*. Available online at: http://www.hajim.rochester.edu/ece/~gsharma/papers/SharmaDarkSideColorEI2012.pdf (accessed May 28, 2019).

Sharma, G., and Trussell, H. J. (1997). Digital color imaging. *IEEE Trans. Image Process.* 6, 901–932. doi: 10.1109/83.597268

Silva, S., Sousa Santos, B., and Madeira, J. (2011). Using color in visualization: a survey. *Comput. Graph.* 35, 320–333. doi: 10.1016/j.cag.2010.11.015

Smith, A. R. (1978). Color gamut transform pairs. *ACM Siggraph Comput. Graph.* 12, 12–19. doi: 10.1145/965139.807361

Smith, N., van der Walt, S., and Firing, E. (2015). *Matplotlib Colormaps*. Available online at: https://matplotlib.org/tutorials/colors/colormaps.html (accessed May 28, 2019).

Stål, T. (2019). *TobbeTripitaka/Grid: First Frame (Version 0.3.0)*. Available online at: http://doi.org/10.5281/zenodo.2553966 (accessed May 28, 2019).

Stauffer, R., Mayr, G. J., Dabernig, M., and Zeileis, A. (2015). Somewhere over the rainbow: how to make effective use of colors in meteorological visualizations. *Bull. Am. Meteorol. Soc.* 96, 203–216. doi: 10.1175/BAMS-D-13-00155.1

Stockman, A., and Brainard, D. H. (2015). "Fundamentals of color vision I: color processing in the eye," in *Handbook of Color Psychology*, eds A. J. Elliot, M. D. Fairchild, and A. Franklin (Cambridge, UK: Cambridge University Press), 27–69. doi: 10.1017/CBO9781107337930.004

Telea, A. C. (2007). *Data Visualization: Principles and Practice*. Boca Raton, FL: AK Peters; CRC Press. doi: 10.1201/b10679

Tkalcic, M., and Tasic, J. F. (2003). "Colour spaces: perceptual, historical and applicational background," in *The IEEE Region 8 EUROCON 2003 Computer as a Tool* (Ljubljana), 304–308.

Tufte, E. R. (1990). *Envisioning Information*. Cheshire, CT: Graphics Press.

Tukey, J. W. (1977). *Exploratory Data Analysis*, Vol. 2. Reading, MA: Addison-Wesley.

Tukey, J. W. (1990). Data-based graphics: visual display in the decades to come. *Stat. Sci.* 5, 327–339. doi: 10.1214/ss/1177012101

Unity Technologies (2019). *Unity*. Available online at: https://unity.com/frontpage

Van Wijk, J. J. (2005). "The value of visualization," in *Visualization, 2005. VIS 05* (Minneapolis, MN: IEEE), 79–86. doi: 10.1109/VISUAL.2005.1532781

Ward, M. O., Grinstein, G., and Keim, D. (2010). *Interactive Data Visualization: Foundations, Techniques, and Applications*. Natick, MA: AK Peters; CRC Press. doi: 10.1201/b10683

Ware, C. (1988). Color sequences for univariate maps: theory, experiments and principles. *IEEE Comput. Graph. Appl.* 8, 41–49. doi: 10.1109/38.7760

Ware, C. (2013). *Information Visualization: Perception for Design, 3rd Edn*. Amsterdam: Elsevier.

Ware, C., Turton, T. L., Bujack, R., Samsel, F., Shrivastava, P., and Rogers, D. H. (2018). Measuring and modeling the feature detection threshold functions of colormaps. *IEEE Trans. Vis. Comput. Graph.* 25, 2777–2790. doi: 10.1109/TVCG.2018.2855742

Wessel, P., Smith, W. H. F., Scharroo, R., Luis, J., and Wobbe, F. (2013). Generic mapping tools: improved version released. *Eos Trans. Am. Geophys. Union* 94, 409–410. doi: 10.1002/2013EO450001

Wickham, H., Cook, D., Hofmann, H., and Buja, A. (2010). Graphical inference for infovis. *IEEE Trans. Visual. Comput. Graph.* 16, 973–979. doi: 10.1109/TVCG.2010.161

Wilkinson, L. (2005). *The Grammar of Graphics, 2nd Edn*. Chicago, IL: Springer Science and Business Media.

Wyszecki, G., and Stiles, W. S. (1982). *Color Science, Vol. 8*. New York, NY: Wiley.

Zeileis, A., and Hornik, K. (2006). *Choosing Color Palettes for Statistical Graphics (No. 41)*. Vienna: Department of Statistics and Mathematics, WU Vienna University of Economics and Business. Retrieved from: http://epub.wu.ac.at/1404/

Zeyen, M., Post, T., Hagen, H., Ahrens, J., Rogers, D., and Bujack, R. (2018). *Color Interpolation for Non-Euclidean Color Spaces*. Berlin: IEEE. doi: 10.1109/SciVis.2018.8823597

Zhou, L., and Hansen, C. D. (2016). A survey of colormaps in visualization. *IEEE Trans. Visual. Comput. Graph.* 22, 2051–2069. doi: 10.1109/TVCG.2015.2489649

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Chapter 6: Exploratory volumetric deep Earth visualization by 2.5D interactive compositing

6 Publication details

A revised version of this chapter has been published as:

Morse, P.E., Reading, A.M., Stal, T., 2020. Exploratory volumetric deep Earth visualization by 2.5D interactive compositing. IEEE Trans. Visual. Comput. Graphics 1–1. https://doi.org/10/gh5n22

Note that thesis page range 91 – 103, are enumerated in journal format text as pp.1-13.

Supplemental material referenced by this publication is appended at the end of this document in the Supplements section.

Online Supplemental materials (including software, movies, images) are available at:

https://doi.org/10.5281/zenodo.3264037

https://github.com/pemorse/data-visualization-tools

Software and code will be also made available at the IEEE digital repository.

Exploratory volumetric deep Earth visualization by 2.5D interactive compositing

Peter E. Morse, Anya M. Reading, and Tobias Stål

Abstract— In this contribution we consider the visualization of global, deep Earth volume datasets for display and researcher interaction. While the algorithms and data analysis techniques that produce such volumetric results have become more sophisticated, the manner of visualizing these findings can be improved. We address the challenge of making an illustrative, exploratory visualization of a global geoscience dataset using a combined seismic tomography result, the primary means by which geoscientists infer structure and process in the deep Earth. We present a novel, interactive graphical application suite and associated workflow that uses an intuitive 2.5D layer compositing approach. This allows the user to adjust the separation between data-slices, control graphics variables such as color mapping, opacity and compositing, and facilitate exploration and annotation of the architecture of the lithosphere. Graphics outputs from our applications are enabled for immersive systems such as dome displays. In a case study we visualize the deep Earth structure beneath the Indian Ocean region. We anticipate that the application methodology will find use in the visualization of multiple datasets representing aspects of the Earth's deep interior and atmosphere, and in the interaction with the increasing number of rich datasets from missions to our neighboring planets.

Index Terms— color mapping, interactive visualization, seismic tomography, solid earth geophysics, volume visualization

1 Introduction

GLOBAL geophysics and planetary sciences face challenging visualization tasks in exploring detailed 2D, 3D and time variant data. Natali et al. [1] observe that rapid modelling and visualization in the early phase of an interpretation process is a key facilitator for elicitation and exchange of expert knowledge, and that this capacity is largely overlooked in the current geoscientific toolset. Rapid light-weight visualization of data for interactive first-pass exploration, analysis and communication facilitates an initial overview of a dataset and can bring significant features to the attention of the analyst.

Deep Earth datasets present special challenges to geophysical visualization. On one hand, widely-used GIS and Virtual Globe toolsets do not readily accommodate subsurface volume data, on the other, Digital Earth and similar specialized software can be difficult to use. A software approach that facilitates interdisciplinary collaboration between visualization expertise and geophysical scientific practices is an appealing proposition.

In this contribution we introduce Planetary Data Tagger Volumetric (PDT_V), an intuitive 2.5D layer-based MacOS application for rapid visualization of planetary scale subsurface geoscientific datasets. Its key objective is to facilitate feature exploration, using interactive 2.5D volumetric visualization and basic metadata generation, for pre-processed deep Earth data via an interactive 3D interface. The animated interface facilitates depth perception of three-dimensional features, with outputs suitable for collaborative display systems.

The volumetric solution presented in this work employs a modified form of texture-slicing, mapping global 2D equirectangular textures to 3D concentric spheres. This is an effective approach for rapid, interactive visualization of subsurface geoscientific data. The layer-based approach will be familiar to users of standard media-production software.

Main Contributions:

- User-friendly interactive visualization of deep Earth structure
- Fast interactive 2.5D volume rendering of subsurface global geophysical data
- Interactive color-mapping via companion apps
- CIELAB/RGB color control via companion apps
- Live video-sharing via Syphon/NDI-enabled outputs for dome displays
- Easily modifiable visual-programming source code

2 Background and Related Work

Geographical Information Systems (GIS) [2], Virtual Globes [3] and Digital Earths [4] provide means of visualizing and analyzing large-scale geospatial and geophysical data [5], [6]. Extending the capabilities and applications of GIS, geovisualization and geovisual analytics [7], [8] arose from developments in digital cartography [9], sharing visualization methodologies in common with Information Visualization (InfoVis), Scientific Visualization (SciVis), and Visual Analytics (VA)[10]–[13]. VA and visualization software for geosciences take the form of desktop applications, specialist libraries and multipart toolchains. These tools require some expertise and focus upon precise model

- *P.E. Morse is with Earth Sciences, Private Bag 79, University of Tasmania, Hobart, TAS 7001, Australia. E-mail: peter.morse@utas.edu.au.*
- *A.M. Reading. is with Physics, Private Bag 37, University of Tasmania, Hobart, TAS 7001, Australia. E-mail:. anya.reading@utas.edu.au*
- *T. Stål is with Earth Sciences and the Institute for Marine and Antarctic Studies, Private Bag 79, University of Tasmania, Hobart, TAS, 7001, Australia E-mail: tobias.staal@utas.edu.au*

value handling and analysis, rather than rapid interactive visualization.

Geospatial visualization, display and analytic systems can take the form of conventional WIMP (Window-Icons-Menus-Pointers) stand-alone applications, web and database-connected solutions, as well as extended reality applications (XR/VR/AR/MR) [14], [15]. Rapid subsurface geoscientific visualization using an intuitive, interactive interface is not well represented in this software category. It presents novel opportunities for preliminary data reconnaissance, with potential extended functionality for screen technologies such as domes [16], [17] and XR-type displays for immersive analytics [18], including interoperability with open source astrographics applications [19], [20].

2.1 3D GISs

Standard desktop GIS tools require extensive training and have not been designed with intuitive exploratory visualization in mind. Some feature subsets of these tools (e.g. basic geovisualization and geotagging [21]) have converged with the advent of cloud-based solutions and APIs [21], [22], WebGL, javascript and HTML5, offering interaction with standard geospatial data abstraction libraries and mapping tools, virtual globe software and visual analytics libraries [23], [24].

GIS systems typically utilize a 2D display, with data drawn to a series of stacked layers registered via georeferenced coordinates and appropriate map projections. Vector and raster data can be overlaid, composited and queried via algorithms built into the software. A characteristic problem for 2D layer approaches to data visualization in GIS is the occlusion and self-occlusion of data, whereby objects in foreground layers obscure underlying ones, or 3D objects in a 2D scene act as visual clutter [25].

3D GIS provide some solutions to this problem through interface designs that allow the rotation and translation of camera position, multiple perspectives, layer blending, sorting, compositing and transparency controls [26]. Interaction considerations, such as means for adjusting scale, orientation, color and style of visual indicators and icons (e.g. glyphs, markers and fields) can also impact visual clarity and interpretation [27].

2.2 Virtual Globes

Virtual Globes provide interactive tools to visualize cartographic and geoscientific data [28]. Google Earth and Google Earth Engine are widely used for geovisualization [29], allowing the incorporation of geoscientific data via open standards such as Python APIs and KML for 2D, basic 3D and animation/navigational data [30]. The COLLADA interchange format is supported for more complex 3D data models [31].

Whilst Google Earth supports vector data (points, lines, polygons), gridded and raster data that can visualize the subsurface, due to limitations in software design they are typically restricted to layers that can only be physically displayed on or above the surface of the Earth [3]. It is currently challenging to move KML objects interactively within the volume of Google Earth to display subsurface features. A common work-around is to represent subsurface phenomena in layers above or intersecting with the Earth's surface [3], [32]. Often, this means draping the data representation upon surface geometry (e.g. a texture layer on a digital elevation model) or cross-section data displayed upon offset or surface-intersecting planes, requiring the user to infer visually the implied spatial relationships. These approaches can create potentially misleading representations of subsurface phenomena.

NASA World Wind [33] and other virtual globes such as Crusta [34], ArcGIS Earth [35] and recent planetary visualization systems [36], similarly focus upon surface features. The opensource World Wind SDK can support subsurface phenomena, also interfacing with java toolkits such as UNAVCO/Unidata IDV [37] that can draw subsurface models.

Subsurface Virtual Globe models could prospectively be constructed using software libraries such as osgEarth [38], OpenGlobe [39], X3D, GeoVRML [40] or three.js, with some programming effort. A recent javascript Digital Globe library, cesium.js, shows great promise in this area [41], and is explicitly used for sophisticated geoscientific subsurface data visualization by the GPlates Portal [42].

2.3 Digital Earth

Most raster-based approaches to geospatial analyses are considered 2.5D, meaning that data (z) varies as a function of position in a regular x-y grid. Commonly (z) is interpreted as elevation, although it could be assigned many other spatial attributes (e.g. density, gravity, magnetics). This effectively precludes full 3D visualization of data in an appropriate x-y-z Cartesian or voxel structure, significantly constraining aspects of geoscientific visualization concerned with volumetric data within the Earth [43].

Digital Earth models attempt to remedy this impasse by explicitly incorporating full 3D coordinate information in Digital Terrain Models (DTM), where elevation data is explicitly encoded, as well assigning possible data values to points (e.g. x,y,z,v) [11]. Subsurface volume data can be explicitly encoded with additional parameters. Examples of this approach can be found in Digital Earth models such as standalone applications SolidEarth [44], ShowEarthModel [45], Terraviz [46], VRML/X3D graphics [47] and comparable 3D KML/KMZ overlays to Google Earth using bespoke graphics [48].

2.4 Volumetric Earth

As a 3D solid it is desirable to be able to represent the Earth volumetrically [49]. Volume data are 3D (possibly time variant) entities that may have indeterminate surfaces or edges that are difficult to isolate or partition geometrically, due to complexity and scale. Typically, volumetric modelling and rendering is computationally expensive, requiring fast CPU/GPU compute and high memory requirements, contingent upon the scale, format and type of data requiring visualization [50].

Direct Volume Rendering (DVR) creates a 2D image directly from 3D volumetric data, using complex raycasting and raymarching algorithms in conjunction with physics-based optical models [51], [52]. Many modern desktop and laptop computers are now capable of managing DVR of

subsampled high-resolution datasets, with online examples implemented in WebGL and javascript [53]. Recent advances in multiresolution architectures enable mobile display applications [54].

Applications for desktop visualization of large datasets such as e.g. Paraview, VisIt, Drishti [55] and GPlates [56], provide rich interfaces for high-end volumetric visualization, including advanced features such as isosurface extraction, scene segmentation, complex transfer function controls and a wide variety of rendering algorithms.

2.5 Layer-based Volume Visualization

Computationally light-weight 2D image-based techniques ('explorable images') for volume visualization have been proposed, using a combination of techniques such as proxy images and intermediate representations, Volumetric Depth Images (VDI) (a generalization of Layered Depth Images (LDI)), and other data attenuation/compression strategies [57], [58]. 2D image-based techniques can often be carried out more efficiently than their 3D counterparts, facilitating interactive exploration.

2.5D compositing/layer-based volume rendering approaches provide a computationally-light alternative to DVR [59] and LDI/VDI. Simplification of the rendering process facilitates interactive performance and provides an intuitive model for image composition [60]. To facilitate this, representations need to control aspects of colorization, shading and transparency in direct response to underlying data values, taking into account aspects of human color perception [61]. Layer-based approaches are standard in media-production software (e.g. Adobe Photoshop, GIMP, Affinity Photo, Natron, Blender etc.) based upon Porter-Duff compositing algebra.

Pre-processed stack-based sub-surface geological representations naturally suggest the utility of similarly efficient forms of volumetric visualization such as layered textures [62]. Pre-processed layers of earth science data can be rapidly visualized using texture-based rendering, rather than raycasting techniques (e.g. DVR), using OpenGL opacity blending, alpha masking and compositing operations.

2.6 Illustration-based Volume Visualization

In addition to photorealistic volume rendering algorithms, non-photorealistic illustration-based techniques have been developed to enhance feature discriminability [63]. 'Illustrative' or feature-highlighting approaches enable visualizations to draw user-attention to significant features, based upon a controllable mixture of stylized photorealism and illustrative abstraction [64]. An illustration-based approach may employ visual methods such as outlining, contrast, color/opacity adjustments and texture variation, amongst others, including interactive techniques such as cut-aways and virtual lenses [65].

3 Visualizing Volumetric Global Earth Data

Seismic tomography volumes are a rich research product that present special problems to volume visualization techniques [66]. Conventional output of multiple colorized seismic depth images in printed and 2D screen display requires that users visually infer the volumetric relationship between single images, e.g. [67], [68]. Alternatively, seismic structures may be visualized in 3D as isosurfaces, potentially creating the visual inference that they are monolithic, whereas in fact they are not. Seismic mantle structures may contain structures-within-structures, demonstrating formal complexity that translucent 3D visualization can elucidate. For any volumetric image representation, the display of semi-transparent structures is an appealing alternative, yet presents both computational and perceptual challenges for effective visualization [69].

The inherent spatial ambiguity of translucent volumetric 3D data on a 2D display may be mitigated by interactive and stereoscopic visualization [70]. However, stereoscopic display systems introduce complexity to the visualization process due to the necessity of rendering both left- and right-eye views. Alternatively, structure-from-motion can be inferred perceptually through interactivity and animation, providing depth-cues through parallax [71].

An application approach that can create easily-reconfigurable interactive volume renderings, including capabilities to control linear color-mapping in perceptually uniform color space (e.g. CIELAB), the ability to interactively modify these color gradients on-the-fly as features of interest are identified, and the ability to visualize these using collaborative animated displays for dome and XR environments does not currently exist in the geoscientific arsenal. Such an approach would form a useful adjunct to advanced specialized subsurface visualization systems such as GPlates [56] and SubMachine [72], as well as astrographics applications like Openspace [19] and Stellarium [73].

4 PDT_V

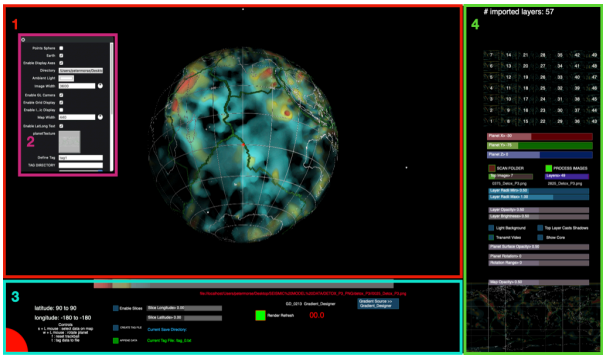


Fig. 1. PDT_V Application GUI (v.0.9.15). High resolution images for all figures are available in Supplement 1.

PDT_V adopts a layer-based approach, common to 2D graphics programs, compositing 2D layers together in a geolocated stack to create an interactive 2.5D volumetric view mapped beneath the Earth's surface. Limited above-surface visualization is also possible. Layer separation is user-adjustable for visualization purposes. The image texture and opacity of the surface layer of the Earth (or another planetary object) can be adjusted to suit the visualization. The application can load a variety of image data file types (PNG, JPEG, TIFF). For global coverage they should be equirectangular images with a 2:1 aspect ratio. Color-

mapping functions require greyscale RGB image data.

The application is programmed in the MacOS Quartz Composer (QC) VPL, using a mixture of pre-defined Objective-C processing nodes as well as custom routines programmed in Objective-C, Javascript and OpenGL. It has been compiled into a stand-alone application, targeting MacOS 10.13 (High Sierra) - 10.15 (Catalina).

4.1 PDT_V User Interface

PDT_V has four main panels (Fig. 1): Panel 1, the main interactive visualization window, displays the current volumetric data layers and current tagged region, mapped to a planetary sphere; Panel 2 (floating parameters window) contains a series of interface controls for changing display elements, data IO and gradient import; Panel 3 contains UI instructions, coordinate information, gradient source selection, tagging controls, pathnames and render refresh controls; Panel 4 displays currently loaded data layers and features compositing, animation, video sharing controls and a map-based region selection interface. The software allows users to specify e.g. speed of earth rotation to facilitate depth perception in the model. Animated movies can be created using screen capture software.

Using cross-linked interactive views, the main window can be zoomed and panned to focus upon features of interest, whilst the number of layers displayed can be adjusted in panel 4 (Fig. 1). Panel 4 can display layers in a stack view or as a grid. Basic lat/long metadata can be generated for regions of interest (ROI). User-defined rectangular ROIs can be selected on the scalable equirectangular map (panel 4), with the selected region drawn live in the main volumetric window using spherical coordinates. Tags are written out as decimal lat/long coordinates and appended to text files created via panel 3. Tags may be assigned arbitrary descriptive strings for user-identification and subsequent processing. Further usage details, high resolution images and movies for the case study examples are provided in Supplement 1.

4.2 Application Usage and Capabilities

PDT_V interfaces with a suite of companion apps (Fig. 2), LAB_CM, GD and GV apps, detailed in [61]. Users can visualize appropriately formatted image data by selecting a folder of sequentially numbered files via the floating parameters windows in both GD and PDT_V. Load time is contingent upon the GPU/CPU, available RAM, number and size of the images and whether scaling is used. The associated workflow is demonstrated in Supplement 1 and the accompanying video.

In PDT_V loaded texture layers will appear wrapped on the Earth sphere at their correct geographical location. Colorization is undertaken via imported image gradients (Fig. 1, panel 2) or live gradients transmitted via Syphon [74] from LAB_CM or GD applications. The LAB_CM app enables perceptually uniform color mapping in CIELAB colorspace (hereafter referenced as 'LAB'), GD provides fine-grained control of RGBa(HSLa) gradient compositing and colorization functions [61].

4.3 Blend Functions, Compositing

PDT_V enables compositing of multiple concentric

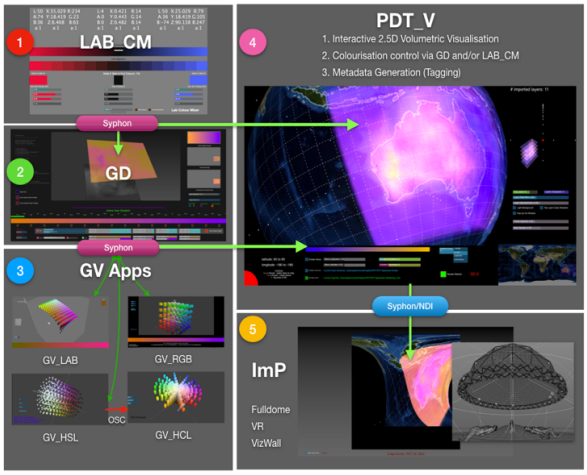


Fig. 2. Application suite workflow: 1) LAB Color Mixer (LAB_CM); 2) Gradient designer (GD); 3) Gradient Visualizer Apps (GV_LAB, GV_RGB, GV_HSL, GV_HCL); 4) Planetary Data Tagger Volumetric (PDT_V); 5) Immersive Player (ImP).

spherical layers, using alpha transparency. This utilizes Porter-Duff compositing (blending) operations implemented in OpenGL, as exposed by QC and underlying MacOS Core Graphics (Quartz) and ColorSync systems. The RGBa blending modes are exposed in PDT_V as "Replace" (no blending), "Over" (pre-multiplied Overlay) and "Add" (source + destination).

In standard operation PDT_V uses the "Over" (pre-multiplied Overlay) blend operation of the basic separable compositing operation [75, pp. 520–522], to form C_r, result color:

$$C_r = \left(1 - \frac{\alpha_s}{\alpha_r}\right) \times C_b + \frac{\alpha_s}{\alpha_r} \times [(1 - \alpha_b) \times C_s + \alpha_b \times B(C_b, C_s)] \quad (1)$$

where C_b = backdrop color, C_s = source color, , α_b = backdrop alpha, α_s = source alpha, α_r = result alpha, $B(C_b, C_s)$ = blend function.

Expressed as a layer function, where a = base layer color, b = top layer color, the Overlay blend operation combines Multiply and Screen blend modes:

$$f(a, \mathrm{b}) = \begin{cases} Multiply\colon (2\mathrm{ab}) & if\ \mathrm{a} < 0.5 \\ Screen\colon (1 - 2(1 - \mathrm{a})(1 - \mathrm{b})) & otherwise \end{cases} \quad (2)$$

Dependent upon the base layer value a, this achieves a linear interpolation between black (a=0), the top layer (a=0.5) and white (a=1). The colors C, a, b are n-element vectors, where n is the number of components required by the colorspace in which compositing is performed, generally being RGBa tuples with normalized values of 0 - 1. LAB and ICCBased spaces cannot function as blending colorspaces, because linear compositing operations in non-linear spaces do not give meaningful results. MacOS and QC automate color management and compositing via a combination of DeviceRGB, sRGB, ICCBased and internal CIEXYZ/CIELAB colorspace representations, performing compositing within the appropriate RGB space.

Using the "Over" blend operation PDT_V achieves a near-linear interpolation between stacked layers, where the top layer is lightened by an underlying light layer and darkened by an underlying dark layer. This is applied across all concentric layers, with an interpolation function that constrains maximal RGBa to white (255,255,255,1). The iterated Overlay function can be adjusted by applied layer parameters to ensure that no clipping occurs.

Within the layer stack a variety of interpolation and compositing relationships can be applied between the alpha value applied by the colorized image-alpha to the layer and the alpha value of the layer applied by PDT_V. By default these are linearly interpolated. LAB_CM and GD control for layer image color and alpha interpolation horizontally across the gradient; PDT_V controls vertically through the concentric 2.5D layer stack. In conjunction with interactive movement, this is sufficient for creating a sense of 3D volumetric depth and a preliminary sense of the shape of structures across the layers.

4.4 Application Display Output

Using the interactive interface, rotational animation can be applied to the planetary sphere. Speed and angle of rotation can be adjusted to provide perceptual depth cues, enabling layer parallax to elicit depth structures and spatial relationships.

PDT_V outputs the live graphical display of panel 1 via an addressable Syphon server. This output can be re-projected in formats suitable for immersive displays such as dome systems using the companion ImP apps (Fig. 2) [76]. Further details are given in Supplement 1.

5 Case Study

5.1 Overview and aims

For a case study we visualize interactively SMEAN2 [77], a global, composite mantle tomography model dating from 2016, and focus on the Indian Ocean region including the continent of Australia. Succeeding SMEAN [67], SMEAN2 is based on S40RTS [78], GyPSUM-S [79] and SAVANI [80] global models. This example concerns the change in a physical property (the speed of seismic S-waves) but datasets of a different nature, for example, magnetic field strength, could equally well be explored.

We aim to demonstrate the capabilities of the application approach to deep Earth and planetary volume visualization, drawing on aspects of human perception for the purpose of identifying features in the volumetric dataset, which may be otherwise difficult to elucidate. Given the nature of the data, non-photorealistic approaches provide a flexible method that may also incorporate aspects of uncertainty in the visualization [81]. These include the use of color interpolation in perceptually uniform colorspace (LAB) as well as linear RGB [61], controllable layer-based compositing operations including alpha channels, feature edge-delineation, outlining and embedding, as well as exploiting stereopsis for depth perception via interactivity and animation.

5.2 Data

The global seismic model SMEAN2 is publicly available from the authors' repository [77]. SMEAN2 is provided in GMT grid format, a variation of netCDF.

Using the software package *agrid* [82], the dataset is sampled at 50km intervals from 25km to 2875km depth, generating 58 concentric equirectangular 3600x1800px 8-bit RGB grayscale PNG images, to generate pixel values from bilinear interpolation of surrounding gridpoints. Values are mapped to 8-bit integer, representing the range from +/- 2.4% perturbation from the Preliminary Reference Earth Model (PREM) S-wave velocity [83]. Values are gated to +/- 2.4% variation from PREM, maintaining a sufficient representation of mantle dynamic range to be mapped to linear RGB grayscale pixel values, as indicated in the Supplement 1, Table 1 (the complete dataset is available at [76]).

The underlying S wavespeed values (a physical property indicative of temperature and composition), in the model slices that form the images, are calculated relative to PREM. Use of relative values enables zones with slower, or faster, than expected seismic wavespeeds to be readily identified without having to take into consideration the general increase in this model parameter with depth. Seismic waves travel more quickly in colder lithosphere and deeper mantle regions [84] which are, by convention in Earth Sciences, displayed using cold colors (blues). Warmer regions are displayed using warm colors (reds). The interplay between changes in seismic wavespeed and factors other than temperature that might influence this parameter, such as composition and grain size, is the subject of ongoing research [85].

In the lithosphere (to ~225km) the range of wavespeed values, indicating considerable heterogeneity in the upper mantle, is wider than it is in the mid and lower mantle. The range in values increases again in the lowermost mantle due to boundary conditions and heat transfer at the outer core (~2885km). To provide sufficient precision, the bitmaps are normalized for +/-2.4% for each layer, maintaining consistent perturbation through the layers. This causes some clipping in upper and lower layers of the model values, where the maximum perturbation exceeds this range.

Our broad region of interest lies under the Indian Ocean, between the continental landmasses of Africa, Antarctica and Australia. We target a global hemisphere view centered upon 75E, 30S near the junction of 4 tectonic plates: Australian, Antarctic, Indian, African. This provides an overview of the Indian Ocean and surrounding continental landmasses including Australia.

We focus on mantle structures in two depth ranges:

1] Lithosphere/Upper mantle (75 - 325km depth)

2] Deep mantle (375 - 2875km depth)

PDT_V normalizes layer radii in an equidistant distribution between 1 and 0, where 1 = Earth Surface (0km), 0 = Earth Centre (6371km). This maps intervening layers to appropriately equidistant concentric spheres, approximating their distribution within the mantle. Normalization settings are indicated in the relevant sections.

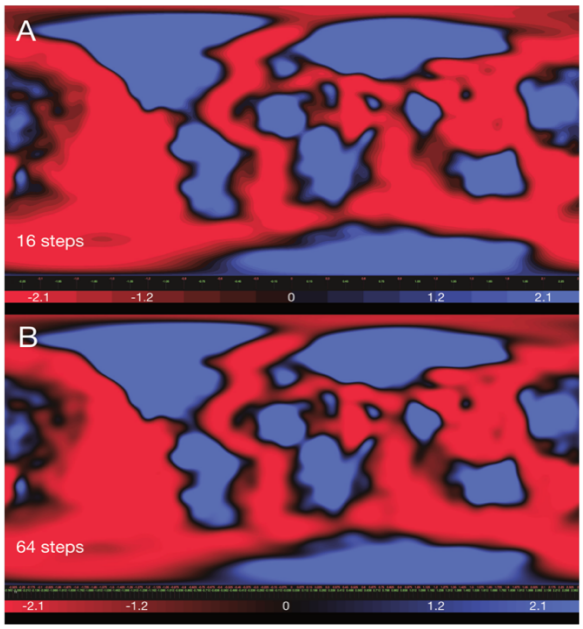


Fig. 3. Effect of gradient quantization upon SMEAN2 data visualization. A = 16 step gradient; B = 64 step gradient. For these single-layer images, value ranges are listed in Supplement 1,Table 2 and shown on the figure scale (+/- % variance from PREM).

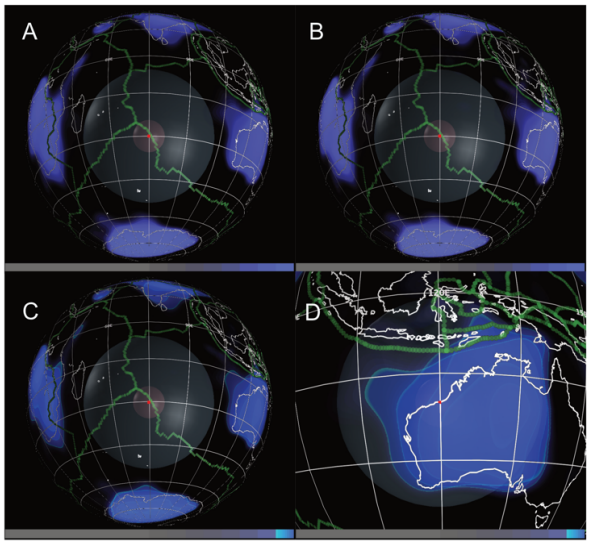


Fig. 4. Illustrative Visualization of the lithosphere/upper mantle in the Indian Ocean region and Australian continent. A and B employ a perceptually uniform colormap, whilst C and D employ a composited colormap. Value ranges are listed in Supplement 1, Table 2. Values are deliberately not shown on the figure itself because the visualizations display multiple composited layers.

5.3 Methods: Colormap and Visualization Workflow

Our initial colormap (Fig. 3) follows a generalized red-blue seismology convention, optimized as per [61]. This colormap is linearly interpolated in perceptually uniform LAB colorspace, produced by the LAB_CM app. We apply this LAB_CM optimized 16-step colormap (Fig. 3A) to the SMEAN2 data. Value ranges are listed in Supplement 1, Table 2. For comparison purposes, Fig. 3B shows the defocus effect of increased quantization from 16 to 64 steps upon the underlying equirectangular-projected SMEAN2 data, visualized using the LAB_CM gradient within GD. For illustrative purposes, we proceed with 16-step quantization for subsequent visualized layers within GD and PDT_V.

Firstly, in Fig. 4 we demonstrate visualization of the lithosphere/upper-mantle, contrasting LAB-based and LAB+RGB(HSL) composited gradients, incorporating illustrative edge outlining. We also note the effects of compositing operations and their relationship with the number of visualized data layers. This figure, and subsequent figures are best viewed as interactive views, videos of rotating spheres, or high resolution images (Supplement 1).

Secondly, for deep mantle visualization, we start with a perceptually-uniform approach (Figs. 5A-D), then proceed to visualization from a perceptual user-angle (Figs. 6A-D), modified using GD compositing operations in RGB(HSL) colorspace, with color-correction and design steps employed to achieve illustrative clarity in the visualization. Following observations in [63], [86], [87] optimizing color choice for volumetric display, we employ a bright, saturated red for the primary target data range, yellow for a secondary target data range, maintaining apparent lightness, whilst providing chromatic distinction. Two more main gradient steps feature near-opponent colors in the cyan-blue region of the HSL spectrum, with reduced apparent lightness, used for 'background' data. Gradients use shading strategies (ranging from bright, saturated colors to semi-transparent black) in order to enhance perception of form, or variable alpha transparency to diminish distinctness, backgrounding the colorized data whilst maintaining visibility of embedded volume structure. Figures are analysed in the following section and the efficacy of this approach is examined in the Discussion.

6 Results

6.1 Upper Mantle Visualization

Fig. 4 demonstrates gradients generated in LAB_CM (LAB colorspace) and GD (RGB colorspace). For comparison purposes, Figs. 4A-B use only LAB_CM gradients, Figs. 4C-D LAB_CM plus GD RGBa(HSLa) gradient overlays. For visualization purposes the Outer and Inner Core visualization feature of PDT_V has been activated. Value ranges are listed in Supplement 1, Table 2.

In Fig. 4A a basic LAB_CM optimized divergent 16-step Red-Black-Blue gradient, with isoluminant termini, and linearly interpolated alpha is applied to 6 image layers, 75km - 325km at 50km intervals, depth-normalized to 0.99-0.95. With low and mid-range value alpha = 0, this shows only fast (continental) features of the upper mantle. Fig. 4B shows 57 layers from 75km – 2875km, depth-normalized to 0.99-0.55, with the same gradient as Fig. 4A. This introduces a perceptible increase in lightness variation introduced by data in layers <325km, even though they are not clearly visible. This suggests that retaining all layers may be important, as they may contribute to the final composite

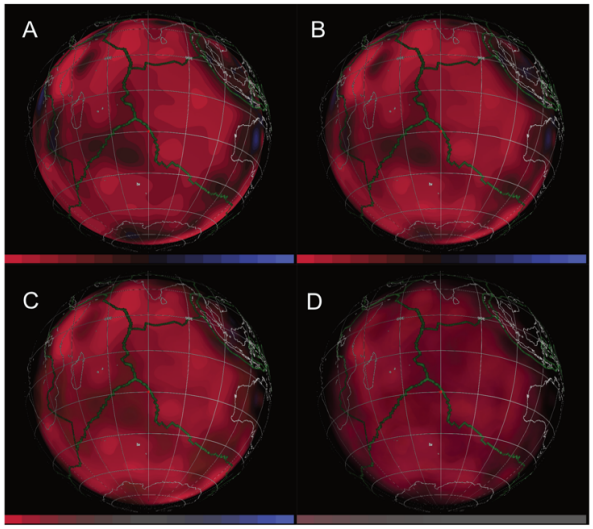


Fig. 5. Visualization of the deep mantle beneath the Indian Ocean region using a perceptually uniform colormap with systematically varying alpha values. See also note on value ranges / multiple layer compositing in the caption to Fig. 4.

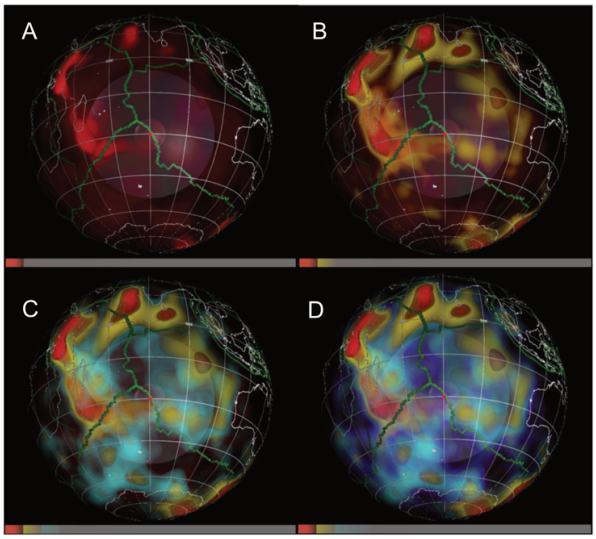


Fig. 6. Visualization of the deep mantle beneath the Indian Ocean region using systematically varying composite gradients. See also note on value ranges / multiple layer compositing in the caption to Fig. 4.

depending upon applied alpha values.

In Fig. 4C GD applies an RGBa(HSLa) overlay gradient to the upper 16th step, highlighting fast lithospheric structure. Maintaining seismology convention for Blue = Fast, an 8-step HSLa gradient ranging from 0.57-0.64 in normalized HSL coordinates is applied, each step a clearly distinguishable 0.1 value (Fig. 4C). Unlike the LAB_CM gradient alone, this introduces clearly defined boundaries to data ranges visualized, which may be desirable given the relatively narrow volumetric constraints of this depth range. Fig. 4D provides a zoomed view of the Australian continent, clearly showing layered feature boundary outlines.

6.2 Deep Earth Perceptually Uniform Gradient

Fig. 5 visualizes the deep mantle from 375km-2875km in 51 image layers at 50km intervals, depth-normalized to 0.94-0.55. Values are listed in Supplement 1, Table 2.

In Fig. 5A, a basic LAB_CM perceptually uniform optimized divergent Red-Black-Blue gradient, with isoluminant termini, and alpha=1 (opaque) is applied to 51 layers of SMEAN2 data displayed in PDT_V, with PDT_V layers alpha = 1. No layers are visible beneath this top layer (375km). This forms our reference image. Fig. 5B maintains LAB_CM gradient alpha = 1. PDT_V layers alpha=0.5 with linear interpolation between layers (*Over* blend function). This begins to reveal underlying layer structure.

Fig. 5C sets LAB_CM central alpha = 0, bottom (Red) and top (Blue) alpha = 0.5. PDT_V layers alpha=0.5 with linear interpolation. This ensures that all layers are partially transparent when composited together, partially revealing obscured / internal structures across the layer stack. Fig. 5D sets LAB_CM bottom (Red) alpha = 0.25, central and top alpha = 0, with PDT_V layers alpha=0.5 with linear interpolation.

Reducing gradient alpha<0.25 demonstrates decreasing capacity to elicit form using alpha compositing based upon the 16 step LAB_CM perceptually uniform gradient. Increasing gradient quantization has negligible visual effect. As transparency linearly interpolates across the gradient, whilst more of the layer stack is visible, the coarseness of the gradient exhibits limited ability to facilitate 3D shape distinction for these more diffuse structures. This indicates that different approaches to gradient design must be employed in 2D and 3D contexts. However, the underlying values, whilst of limited immediate visual use, may be used to help enhance shape distinction with GD applied gradients, with non-visible data affecting visibility up and down the layer stack.

6.3 Deep Earth Composite Gradients

In Figs. 6A-D we take advantage of the ability of GD to composite gradients together. The LAB_CM gradient is activated in the LAB layer of GD, with divergent gradient alpha = 0.1, 0, 0. The QC colorspace render mode for the GD LAB layer is 'linear' in order to maintain linear mapping between its values and the black and white linear RGB data in the SMEAN2 data files. Downstream, this is color-corrected for screen display by QC, without changing any of the underlying linear mapping, enabling correct compositing of RGB values through GD's HSL color application interface. Gradient color values are listed in Supplement 1, Table 2.

In Fig. 6A GD is used to apply an 8-step gradient overlay to the first step of the 16 step LAB_CM gradient, with a bright red to black HSL gradient (S1), with alpha =0.5. This outlines the lowest wavespeed structures in the mantle, drawing on color choices optimized for volumetric display.

In Fig. 6B a second GD gradient (S2) is applied over the second LAB_CM step. This increments to H=0.15 in normalized HSL coordinates, generating a bright yellow. L_{HSL} is set to 0.21, in order to match L_{LAB} for S1 as displayed in GV_LAB, approximating isoluminance [61]. Alpha is

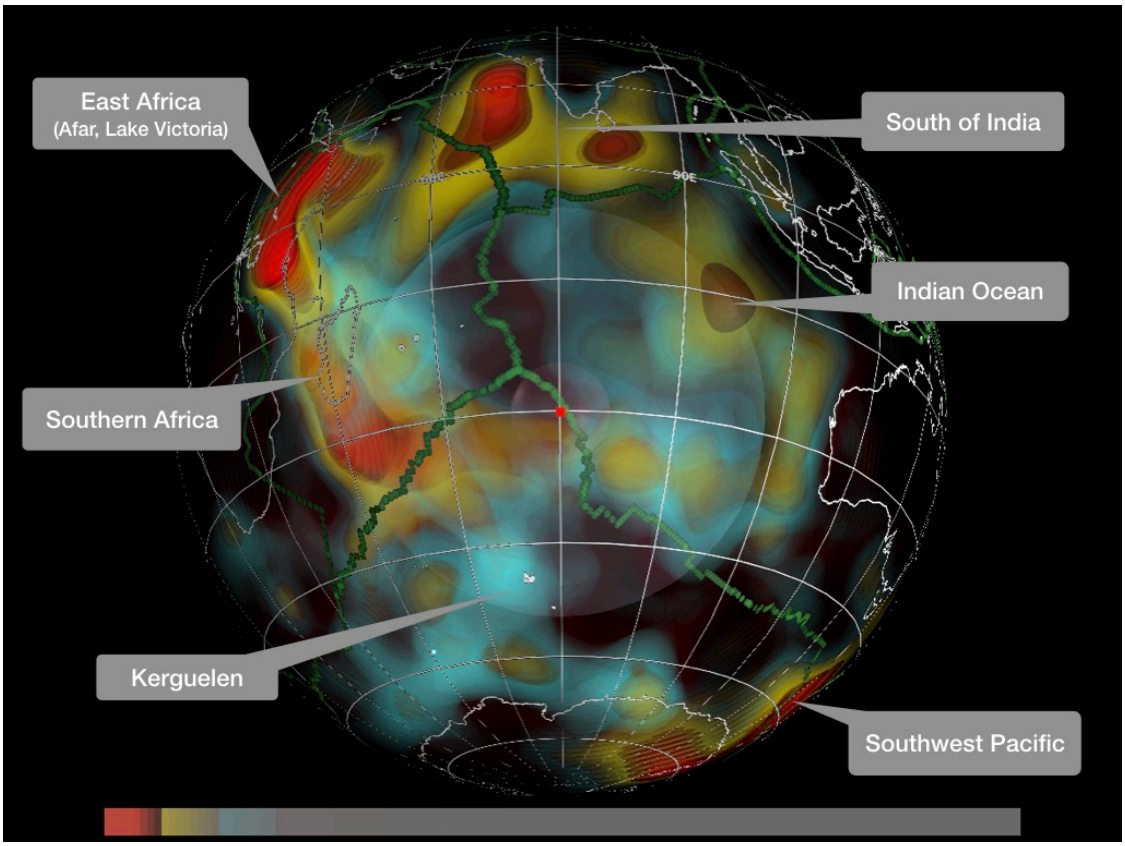


Fig. 7. Exploratory visualization of the deep mantle beneath the Indian Ocean region using the preferred composite gradient (as in Fig.6C, features of interest are identified). See also note on value ranges / multiple layer compositing in the caption to Fig. 4.

linearly interpolated across the S2 gradient from 0.5 to 0. This ensures the identified features fade out at the edges and reveal internal embedded structures delineated in S1.

In Fig. 6C a third GD gradient (S3) is applied over the third LAB_CM step. This is set to H=0.5 (cyan) in normalized HSL coordinates. Again, L_{HSL} is set to 0.21, in order to match L_{LAB} for S1 in GV_LAB, approximating isoluminance. Alpha is linearly interpolated across the gradient from 0.5 to 0. This also ensures the identified features fade out at the edges and reveal internal embedded structures delineated in S1 and S2.

In Fig. 6D a fourth gradient (S4) is applied over the fourth LAB_CM step. This is set to H=0.6 (blue) in normalized HSL coordinates. L_{HSL} is set to 0.1 in order to minimize brightening effects driven by underlaying colorized data values in the volume, whilst still imparting useful colorization to data. Alpha is linearly interpolated across the gradient from 0.05 to 0. This also ensures the identified features fade out at the edges and reveal internal embedded structures delineated by S1, S2 and S3.

6.4 Identified Features

Using the composited gradient developed in Fig. 6C, we are now in a position to explore slow wavespeed regions in our case study visualization, optimized for volumetric feature exploration. While the features are visible on the enlargement in Fig. 7, viewing the animated, rotating form is recommended to draw on human stereopsis and hence make best use of the composited visualization [76].

The most evident slow feature is the deep-seated mantle anomaly beneath east Africa, the form of which may be seen in the upper mantle (down to 600 km) beneath the Afar and Lake Victoria region, connected to an extended feature in the lower mantle (deeper than 800 km) beneath southern Africa [80], [88]. An extensive upper mantle feature in the deeper part of the upper mantle with a more subtle wavespeed anomaly is evident to the west and southwest of the Kerguelen region beneath the Antarctic Plate. South of India, laterally extensive upper mantle features are evident as volumes with much more limited depth extent than the Afar feature, however feature components in the mid-mantle have only a moderate wavespeed anomaly (Supplement 1, Table 2) and are deemphasized by the choices we have made in this suite of visualizations. The northeast Indian Ocean feature, and other features to the west and southwest of Australia may be seen to have a restricted depth extent in the lower part of the upper mantle. Slow features in the southwest Pacific, with considerable depth extent through the upper mantle are also evident (edge of globe in Fig. 7). Features in the lowermost mantle have attracted recent attention as 'Earth Blobs' [89], [90] and this visualization methodology will be

a useful tool to progress this active area of planetary science.

Animation of the visualization enables far better spatial understanding of the structures revealed in the visualization, clearly illustrating the depth relationships between structures and their disposition with global deep Earth structure. Links are given in Supplement 1.

7 Discussion

Our visualizations undertook two complementary structured approaches: a numerical optimization within a perceptual colorspace (LAB+alpha) [61] and a perceptual optimization within a numerical colorspace (RGBa/HSLa) [87]. These were applied to the visualization of upper and lower mantle structures.

Firstly, using LAB_CM, we created a 3-color divergent Red-Black-Blue gradient, broadly following geoscientific convention for seismic imaging [61]. Unlike a conventional non-linear RGB gradient, this was explicitly created and optimized in perceptually uniform LAB colorspace in order to create a perceptually linear gradient, with isoluminant termini and linear ΔE (Euclidean color difference in LAB colorspace) matching the underlying data values. The linearity of the gradient geometry was visualized in the companion GV_LAB app (Fig. 2).

Using the LAB_CM gradient we undertook an initial reconnaissance of the SMEAN2 dataset in PDT_V, focusing upon the 75km-325km depth range, highlighting upper-mantle continental features in blue (Figs. 4A-B). This established the adequacy of a 16-step gradient for visualizing these features in 6 data slices. This gradient was replicated and refined in GD (Fig. 4C), clearly demonstrating the outlining function for shallow depth layers in Fig. 4D.

To visualize deep mantle structures, the LAB_CM gradient was then applied to 51 data layers in PDT_V (Fig. 5), with PDT alpha = 1, displaying only the topmost opaque layer (Fig. 5A). In Fig. 5B we set PDT_V alpha to 0.5, which applies uniformly to all layers displayed in the application. With the Overlay blend mode applied, this ensured a linear interpolation between the 51 concentric layers, such that no clipping occurred. The impression of depth structure began to emerge, but was self-occluding due to the visibility of the entire gradient range. Note that the application of alpha and applied blending mode do not provide meaningful results within LAB colorspace (consequently invalidating ΔE as a meaningful metric at this point). However, as the blending colorspace is managed by QC and the OS, this modified gradient can be usefully visualized within GV_LAB to monitor and approximate linearity, isoluminance and perceptual uniformity [61].

In Fig. 5B we set the gradient center alpha = 0, effectively 'knocking out' the black color, with linear alpha interpolation from 0 – 1 proceeding from the center to the respective red and blue gradient termini. As the gradient consisted of 8 steps either side of the central value (16 steps all together), Δalpha = 0.125 per step. At this point very little blue (=fast) structure remained visible, due to the infrequency of this type of structure in this region of the Earth below the 375km layer, according to our data model (SMEAN2).

In Fig. 5C we set the gradient termini alpha = 0.5. In concert with PDT_V alpha = 0.5, this reduced the apparent PDT_V display alpha = 0.25, rendering the concentric spheres substantially transparent. This had the effect of revealing volumetric structure, most noticeably upon the limbs of the Earth. In Fig. 5D we set the blue terminus alpha = 0, eliminating any representation of > median PREM or SMEAN2 values (faster, denser mantle structure). This made little difference to the visualization, as expected, based upon results in Fig. 5B-C. Fig. 5D also demonstrates the visual effect of decreasing red terminus alpha to 0.25.

Whilst a mild increase in the sense of volumetric depth is obtained through the increasing transparency of the structure, loss of dynamic range (vividness) in color and lightness/luminosity provides decreasing returns on the ability to distinguish structure within a statically visualized dataset. This can be mitigated by turning on the animation functions of PDT_V or by interactively rotating the dataset.

At this point we have approached the useful limit of this type of numerical approach to the visualization, based upon the initial parameters of the 16-step applied LAB_CM gradient with linear alpha interpolation across the gradient and vertically through the composited layers. We have explored, in a controlled linear fashion, a region of the large parameter space made available by the software. The established underlying color and transparency structure is then used to modify (lighten/darken) perceptually optimized gradients overlaid using GD, working in RGBa(HSLa) colorspace.

In Fig. 6 the inner and outer core are visible, clearly revealing the transparency of an applied LAB_CM gradient with an alpha range of 0.1-0. Using Syphon, LAB_CM pipes linear gradient data to GD, GD pipes to PDT_V. The GD LAB layer is set to 'linear' mode (= no automatic QC color correction applied), so that LAB color values are linearly applied to the linear RGB grayscale values of the SMEAN2 image files. However, these are color corrected for display onscreen using internal QC/OS color correction functions that automatically adapt for the display environment (e.g. sRGB/ICC Profile). GD can then apply linear RGB gradients via the HSL slider UI, in an OS-managed sRGB/ICC corrected display colorspace. Here the interplay between perceptually linear LAB colors (LAB_CM), numerically linear RGB/HSL colors (GD) and color-corrected screen colors (OS + hardware) become complex and require monitoring through the GV_LAB app, which can be set to monitor the output of GD over Syphon.

With this arrangement, GD was set to a 16-step gradient, with each step quantized to 8 sub-units interpolated in RGB colorspace. In Fig. 6A, a bright red color (H = 0) was applied interpolating from red (UC1) to black (UC2) overlaying LAB_CM step 1 (S1), with a constant alpha of 0.5. This immediately highlighted the lowest values in the data, applying the visual appearance of a bright central core with a shaded outer perimeter, for the strongest signals in the data, whilst still maintaining a degree of internal translucency.

In Fig. 6B, an additional gradient (S2) is applied over the

second LAB_CM step. Departing from seismology convention, this increments to H=0.15 in normalized HSL coordinates, generating a bright yellow. The yellow provides a striking perceptual contrast to red, whilst remaining proximate in HSL coordinates. L_{HSL} is set to 0.21, in order to match L_{LAB} for S1 as displayed in GV_LAB, approximating isoluminance. Alpha is linearly interpolated across the S2 gradient from 0.5 to 0. This ensures the identified features fade out at the edges and reveal internal embedded structures delineated in S1.

In Fig. 6C a third gradient (S3) is applied over the LAB_CM step 3. This is set to H=0.5 (Cyan) in normalized HSL coordinates. Cyan is approximately opponent to yellow in HSL coordinates, providing strong chromatic contrast. Again, L_{HSL} is set to 0.21, in order to match L_{LAB} for S1 in GV_LAB, approximating isoluminance. Alpha is linearly interpolated across the gradient from 0.5 to 0. This also ensures the identified features fade out at the edges and reveal internal embedded structures delineated in S1 and S2.

In Fig. 6D a fourth gradient (S4) is applied over the LAB_CM step 4. Again departing from standard seismology red-blue convention for the entire data range, this is set to H=0.6 (blue) in normalized HSL coordinates for this subset of the data. Blue is proximate to cyan in HSL coordinates, providing less direct chromatic contrast, whilst still providing a framing/outlining structure to the denser cyan regions. L_{HSL} is set to 0.1 in order to minimize brightening effects driven by underlaying colorized data values in the volume, whilst imparting useful colorization to data. Alpha is linearly interpolated across the gradient from 0.05 to 0. This also ensures the identified features fade out at the edges and reveal internal embedded structures delineated by S1, S2 and S3.

It is pertinent here to consider the data being visualized: these parameters were chosen following a strategy of convention, numerical optimization and interactive exploration based upon knowledge of human visual perception and prior research in volumetric color selection [86], [87]. With these parameters we have arrived at a first-pass numerically-justified and perceptually judicious visualization, concordant with geophysical knowledge exploration. The parameter-space is large, and many permutations are possible, potentially leading to a wide variety of visualization results. The task this presents to geoscientific visualization is to establish, quantify and qualify the relationship between the space of possible representations and those subsets that exhibit inferential justification and utility. This includes a nuanced awareness of the different requirements for 2D versus 3D data colorization, with particular attention to the challenges presented by visualizing features as semi-transparent structures [91].

7.1 Limitations

Our software was implemented in the QC VPL, using some custom routines programmed in Objective-C, OpenGL, GLSL and Javascript. Applications were compiled using QuartzBuilder and XCode. QC was chosen for its graphical programming paradigm, live code performance and wide API support. Since 2018, QC and OpenGL have been deprecated by Apple, though our software has been successfully tested upon MacOS 10.13 (High Sierra) - 10.15 (Catalina). Near-future development options for the application suite include porting to Metal or reimplementation via alternative closed-source (e.g. Unity, Touch Designer) or opensource (e.g. Vuo, Godot, Unreal) frameworks for cross-platform support. These environments can leverage graphics hardware and programming advances from the games and entertainment industries for interactive scientific visualization.

A key limitation in our software is a 'feature' of QC – automatic color-correction hand-off between programming environment and OS that is designed to simplify and semi-automate color-management. In our research this has required careful handling, as it has been difficult to control precisely and monitor the color-mapping and compositing pipeline from LAB/RGB/HSL to sRGB/ICC Profile/Device colorspaces and thence to screen display: these are better handled via explicit graphics programming. More precise control at a fundamental programmatic level than an environment like QC affords will be a factor in the choice of development options noted above. This will result in better-handled modes of alpha compositing, apparent depth relationships and the color-shifts that may incur. A second limitation is introduced by the use of 2.5D compositing: this restricts the viewing angles at which a volumetric effect is achieved, unlike a DVR approach. However, this is mitigated by the low computational overhead of drawing 2D layers to screen rather than voxels, and the concomitant increase in interactivity. This enables our software to employ inactive animation on desktop computers, exploiting the powerful depth effects of stereopsis via motion parallax.

7.2 Interaction with Submachine

A set of tools, SubMachine [72] has been developed by Earth Sciences specialists to facilitate the exploration of deep Earth structure, with an emphasis on seismic tomography, as in our current contribution. Submachine provides a web-based toolset for the interactive 2D visualization of quantitative slices and cross-sections through a wide variety of volumetric deep Earth datasets and models, as well as comparisons between models, plotting and calculation functions (Supplement 1 provides a further example using data from this source). Our new software application aims to complement initiatives of this kind, providing an alternative approach to reconnaissance exploration of large Earth datasets. Submachine results can be exported in appropriate projection formats, cropped and color-processed in preparation for display with GD and PDT_V. This process could be automated with some programming effort. An advantage of this approach would be to interface PDT_V with a pre-existing web-service dedicated to subsurface data. PDT_V provides an interactive, animatable 2.5D volumetric visualization interface and the ability to display these results upon immersive visualization systems. These different approaches to interrogating and displaying data and visualization may advantage collaboration across a range of platforms and disciplines, enabling synergies in the exploration of the deep Earth and

other planetary bodies.

7.3 ImP, Future Work

PDT_V can transmit video frames to external Syphon-capable applications. A demonstration display application, ImP (Fig. 2, panel 5) reads incoming PDT_V Syphon video data and applies appropriate warping for immersive display systems such as dome projection [92]. This is appealing for collaborative multi-participant visualization and public engagement. Implementations for dome and XR are under development [76], as shown in Supplement 1 and the accompanying video. Future developments will extend the 8-bit pipeline to 16- and 32-bit color and alpha compositing operations, and explore the use of industry-standard filetypes such as OpenEXR, which may encapsulate a wider range of data types useful for sophisticated visualization approaches, including stereoscopic depth-mapping, 3-dimensional metadata embedding, masking, focus and occlusion control. These also include high-dynamic range visualization, capable of capturing and expressing the wide dynamic range of seismic data and rendering it to wide gamut colorspaces, for presentation and interaction on next-generation display systems.

8 Conclusion

We have demonstrated the utility of an intuitive interactive 2.5D layer-based approach to visualize interactively Earth mantle structure. Two complementary approaches, numerical and perceptual, were employed, enabling the creation and assessment of a range of adaptable color gradients. This revealed the different applicability of color gradients for 2D and 3D visualization, with particular reference to the use of alpha transparency in compositing for 2.5D volumetric visualization. We demonstrated the utility of 2.5D compositing, interactivity and animation to reveal structures in seismic volume visualization, consistent with, and improving on, existing interpretations of seismic tomographic data. In contrast to static image results, our approach reveals internal structure to volumetric entities, rather than apparently monolithic isosurfaces, and provides strong spatial relationship cues between features. The visualizations are suitable for display upon desktop systems and current immersive screen technologies

These are novel tools for geoscientific visualization, providing a new capacity in data reconnaissance and analysis through efficient interactive, illustrative volume visualization in concert with control of colorization parameters that maintain contiguity with underlying data sources. Computational demands are minimized through the layer-based approach, maintaining interactive explorability of the dataset by alleviating display-machine data-processing overheads via network-accessible data sources. Further development of these approaches, in concert with pre-existing web-services and more numerous volumetric datasets of Earth and other planetary bodies, is set to make data exploration richer and more revealing, and immersive visualization accessible to wider specialist and interdisciplinary audiences.

Acknowledgments

PM co-designed the study, developed the visualization software and wrote the text. AR co-designed the study and contributed to the text. TS carried out data pre-processing and contributed to the text. The authors thank Stephen J. Walters for constructive feedback. PM acknowledges an RTP Scholarship from the University of Tasmania. TS acknowledges support under Australian Research Council's Special Research Initiative for Antarctic Gateway Partnership (Project ID SR140300001). This research was partly supported by the ARC Research Hub for Transforming the Mining Value Chain (project number IH130200004).

References

[1] M. Natali, E. M. Lidal, J. Parulek, I. Viola, and D. Patel, "Modeling Terrains and Subsurface Geology," *Eurographics 2013 - State of the Art Reports*, p. 19 pages, 2012, doi: 10/ggxxtx.

[2] K.-T. Chang, *Introduction to geographic information systems*, Ninth Edition. New York: McGraw-Hill Education, 2018.

[3] L. Yu and P. Gong, "Google Earth as a virtual globe tool for Earth science applications at the global scale: progress and perspectives," *International Journal of Remote Sensing*, vol. 33, no. 12, pp. 3966–3986, Jun. 2012, doi: 10.1080/01431161.2011.636081.

[4] M. A. Brovelli, M. Minghini, R. Moreno-Sanchez, and R. Oliveira, "Free and open source software for geospatial applications (FOSS4G) to support Future Earth," *International Journal of Digital Earth*, vol. 10, no. 4, pp. 386–404, Apr. 2017, doi: 10.1080/17538947.2016.1196505.

[5] C. M. R. Fowler, *The solid earth: an introduction to global geophysics*, 2nd ed. Cambridge University Press, 2008.

[6] M. O. Ward, G. Grinstein, and D. Keim, *Interactive data visualization: foundations, techniques, and applications*. Natick, Massachusetts: AK Peters/CRC Press, 2010.

[7] G. Andrienko, N. Andrienko, D. Keim, A. M. MacEachren, and S. Wrobel, "Challenging problems of geospatial visual analytics," *Journal of Visual Languages & Computing*, vol. 22, no. 4, pp. 251–256, Aug. 2011, doi: 10.1016/j.jvlc.2011.04.001.

[8] A. Çöltekin, S. Bleisch, G. Andrienko, and J. Dykes, "Persistent challenges in geovisualization – a community perspective," *International Journal of Cartography*, vol. 3, no. sup1, pp. 115–139, Oct. 2017, doi: 10.1080/23729333.2017.1302910.

[9] A. M. MacEachren and M.-J. Kraak, "Research Challenges in Geovisualization," *Cartography and Geographic Information Science*, vol. 28, no. 1, pp. 3–12, Jan. 2001, doi: 10.1559/152304001782173970.

[10] J. Dill, R. Earnshaw, D. Kasik, J. Vince, and P. C. Wong, Eds., *Expanding the Frontiers of Visual Analytics and Visualization*. London: Springer London, 2012.

[11] C. D. Hansen and C. R. Johnson, Eds., *The visualization handbook*. Amsterdam ; Boston: Elsevier-Butterworth Heinemann, 2005.

[12] T. Munzner, "A Nested Model for Visualization Design and Validation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 921–928, Nov. 2009, doi: 10.1109/TVCG.2009.111.

[13] M. Nöllenburg, "Geographic visualization," in *Human-Centered Visualization Environments*, A. Kerren, Ed. Berlin Heidelberg: Springer, 2007, pp. 257–294.

[14] D. W. F. Van Krevelen and R. Poelman, "A survey of augmented reality technologies, applications and limitations," *International journal of virtual reality*, vol. 9, no. 2, pp. 1–20, 2010, doi: 10/ggxxt5.

[15] J. Wilford, "Scientific visualisation and 3D modelling applications for mineral exploration and environmental management," *AGSO Research Newsletter*, vol. 31, 1999, Accessed: Jun. 02, 2020. [Online]. Available: http://crcleme.org.au/Research/P4_SalinityMappingMethods04/abstracts/20o_Wilford/wilford_scientific_visualisation_mineral_exploration_envi.pdf.

[16] P. Bourke, "iDome: Immersive gaming with the Unity3D game engine," *Computer Games and Allied Technology*, vol. 9, 2009, doi: 10/ggxxvb.

[17] S. Li *et al.*, "Interactive theater-sized dome design for edutainment and immersive training," in *Proceedings of the 2014 Virtual Reality International Conference*, 2014, p. 8, doi: 10/gf8z7r.

[18] K. Marriott *et al.*, "Immersive Analytics: Time to Reconsider the Value of 3D for Information Visualisation," in *Immersive Analytics*, vol. 11190, K. Marriott, F. Schreiber, T. Dwyer, K. Klein, N. H. Riche, T. Itoh, W. Stuerzlinger, and B. H. Thomas, Eds. Springer International Publishing, 2018, pp. 25–55.
[19] A. Bock *et al.*, "OpenSpace: A System for Astrographics," *IEEE Trans. Visual. Comput. Graphics*, pp. 1–1, 2019, doi: 10/ggsrbm.
[20] M. G. Connors and B. Martin, "Stellarium as an Educational Resource in Planetary Astronomy," *AGU Fall Meeting Abstracts*, vol. 54, pp. ED54A-08, Dec. 2009.
[21] S. Elwood, "Geographic Information Science: new geovisualization technologies — emerging questions and linkages with GIScience research," *Progress in Human Geography*, vol. 33, no. 2, pp. 256–263, Apr. 2009, doi: 10.1177/0309132508094076.
[22] Q. Huang *et al.*, "Evaluating open-source cloud computing solutions for geosciences," *Computers & Geosciences*, vol. 59, pp. 41–52, Sep. 2013, doi: 10.1016/j.cageo.2013.05.001.
[23] M. Bostock, V. Ogievetsky, and J. Heer, "D^3 Data-Driven Documents," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2301–2309, Dec. 2011, doi: 10.1109/TVCG.2011.185.
[24] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer, "Vega-Lite: A Grammar of Interactive Graphics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 341–350, Jan. 2017, doi: 10.1109/TVCG.2016.2599030.
[25] S. Brooks and J. L. Whalley, "Multilayer hybrid visualizations to support 3D GIS," *Computers, Environment and Urban Systems*, vol. 32, no. 4, pp. 278–292, Jul. 2008, doi: 10.1016/j.compenvurbsys.2007.11.001.
[26] S. Dubel, M. Rohlig, H. Schumann, and M. Trapp, "2D and 3D presentation of spatial data: A systematic review," in *2014 IEEE VIS International Workshop on 3DVis (3DVis)*, Paris, France, Nov. 2014, pp. 11–18, doi: 10.1109/3DVis.2014.7160094.
[27] M. M. Haklay, *Interacting with geospatial technologies*. John Wiley & Sons, 2010.
[28] J. E. Bailey and A. Chen, "The role of Virtual Globes in geoscience," *Computers & Geosciences*, vol. 37, no. 1, pp. 1–2, Jan. 2011, doi: 10.1016/j.cageo.2010.06.001.
[29] J. Liang, J. Gong, and W. Li, "Applications and impacts of Google Earth: A decadal review (2006–2016)," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 146, pp. 91–107, Dec. 2018, doi: 10.1016/j.isprsjprs.2018.08.019.
[30] L. M. Ballagh *et al.*, "Representing scientific data sets in KML: Methods and challenges," *Computers & Geosciences*, vol. 37, no. 1, pp. 57–64, Jan. 2011, doi: 10.1016/j.cageo.2010.05.004.
[31] D. G. De Paor and S. J. Whitmeyer, "Geological and geophysical modeling on virtual globes using KML, COLLADA, and Javascript," *Computers & Geosciences*, vol. 37, no. 1, pp. 100–110, Jan. 2011, doi: 10.1016/j.cageo.2010.05.003.
[32] Y. Yamagishi *et al.*, "Visualization of geoscience data on Google Earth: Development of a data converter system for seismic tomographic models," *Computers & Geosciences*, vol. 36, no. 3, pp. 373–382, Mar. 2010, doi: 10.1016/j.cageo.2009.08.007.
[33] D. G. Bell *et al.*, "NASA World Wind: Opensource GIS for Mission Operations," in *2007 IEEE Aerospace Conference*, Big Sky, MT, USA, 2007, pp. 1–9, doi: 10.1109/AERO.2007.352954.
[34] T. Bernardin *et al.*, "Crusta: A new virtual globe for real-time visualization of sub-meter digital topography at planetary scales," *Computers & Geosciences*, vol. 37, no. 1, pp. 75–85, Jan. 2011, doi: 10.1016/j.cageo.2010.02.006.
[35] ESRI, "ArcGIS Earth," 2019. https://www.esri.com/en-us/arcgis/products/arcgis-earth (accessed Jan. 21, 2020).
[36] K. Bladin *et al.*, "Globe Browsing: Contextualized Spatio-Temporal Planetary Surface Visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 802–811, Jan. 2018, doi: 10/gcqbjw.
[37] D. Murray, J. McWhirter, Y. Ho, and T. Whittaker, "The IDV At 5: New Features and Future Plans," in *Proceedings of 25th Conference on International Interactive Information and Processing Systems (IIPS) for Meteorology, Oceanography, and Hydrology*, 2009, vol. 7, p. 2 pages, Accessed: Jun. 02, 2020. [Online]. Available: https://ams.confex.com/ams/pdfpapers/150736.pdf.
[38] Q. Chen, G. Liu, X. Ma, Z. Yao, Y. Tian, and H. Wang, "A virtual globe-based integration and visualization framework for aboveground and underground 3D spatial objects," *Earth Science Informatics*, vol. 11, no. 4, pp. 591–603, Dec. 2018, doi: 10.1007/s12145-018-0350-x.
[39] P. Cozzi and K. Ring, *3D Engine Design for Virtual Globes*, 1st ed. USA: A. K. Peters, Ltd., 2011.
[40] M. P. McCann, "Using GeoVRML for 3D oceanographic data visualizations," in *Proceedings of the ninth international conference on 3D Web technology - Web3D '04*, Monterey, California, 2004, p. 15, doi: 10.1145/985040.985043.
[41] Evangelidis, Konstantinos; Papadopoulus, Theofilos, "Is there life in Virtual Globes?," in *Free and Open Source Software for Geospatial (FOSS4G), Conference Proceedings*, Bonn, Germany, 2016, vol. 16, pp. 50–55, doi: 10.7275/r56m3518.
[42] R. D. Müller *et al.*, "The GPlates Portal: Cloud-Based Interactive 3D Visualization of Global Geophysical and Geological Data in a Web Browser," *PLOS ONE*, vol. 11, no. 3, p. e0150883, Mar. 2016, doi: 10.1371/journal.pone.0150883.
[43] T. F. Wawrzyniec, R. R. Jones, K. McCaffrey, J. Imber, N. Holliman, and R. E. Holdsworth, "Introduction: Unlocking 3D earth systems—Harnessing new digital technologies to revolutionize multi-scale geological models," *Geosphere*, vol. 3, no. 6, p. 406, 2007, doi: 10.1130/GES00156.1.
[44] L. Zhu, J. Sun, C. Li, and B. Zhang, "SolidEarth: a new Digital Earth system for the modeling and visualization of the whole Earth space," *Frontiers of Earth Science*, vol. 8, no. 4, pp. 524–539, Dec. 2014, doi: 10.1007/s11707-014-0438-7.
[45] M. A. Jadamec, O. Kreylos, B. Chang, K. M. Fischer, and M. B. Yikilmaz, "A Visual Survey of Global Slab Geometries With ShowEarth-Model and Implications for a Three-Dimensional Subduction Paradigm," *Earth and Space Science*, vol. 5, no. 6, pp. 240–257, 2018, doi: 10/gfn5cs.
[46] J. W. Stoecker, "TerraVis: A Stereoscopic Viewer for Interactive Seismic Data Visualization," Ph.D. Thesis, University of Miami, 2011.
[47] A. Plesch and M. McCann, "The X3D geospatial component: X3DOM implementation of GeoOrigin, GeoLocation, GeoViewpoint, and GeoPositionInterpolator nodes," in *Proceedings of the 20th International Conference on 3D Web Technology - Web3D '15*, Heraklion, Crete, Greece, 2015, pp. 31–37, doi: 10.1145/2775292.2775315.
[48] J. R. Ziolkowska and R. Reyes, "Geological and hydrological visualization models for Digital Earth representation," *Computers & Geosciences*, vol. 94, pp. 31–39, Sep. 2016, doi: 10.1016/j.cageo.2016.06.003.
[49] A. K. Turner, "Challenges and trends for geological modelling and visualisation," *Bulletin of Engineering Geology and the Environment*, vol. 65, no. 2, pp. 109–127, May 2006, doi: 10.1007/s10064-005-0015-0.
[50] R. Westermann and T. Ertl, "Efficiently using graphics hardware in volume rendering applications," in *Proceedings of the 25th annual conference on Computer graphics and interactive techniques - SIGGRAPH '98*, 1998, pp. 169–177, doi: 10/df7z5s.
[51] A. Kaufman and K. Mueller, "Overview of Volume Rendering," *Visualization Handbook*, pp. 127–174, 2005, doi: 10.1016/B978-012387582-2/50009-5.
[52] K. Engel, M. Hadwiger, J. M. Kniss, A. E. Lefohn, C. R. Salama, and D. Weiskopf, "Real-Time Volume Graphics," *ACM Siggraph 2004 Course Notes*, p. 282 Pages, 2004, doi: 10/bbk4hs.
[53] M. M. Mobeen and L. Feng, "High-Performance Volume Rendering on the Ubiquitous WebGL Platform," in *2012 IEEE 14th International Conference on High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems*, Liverpool, United Kingdom, Jun. 2012, pp. 381–388, doi: 10.1109/HPCC.2012.58.
[54] F. Mwalongo, M. Krone, G. Reina, and T. Ertl, "State-of-the-Art Report in Web-based Visualization," *Computer Graphics Forum*, vol. 35, no. 3, pp. 553–575, Jun. 2016, doi: 10/f8vxfs.
[55] A. Limaye, "Drishti: a volume exploration and presentation tool," San Diego, California, USA, Oct. 2012, p. 85060X, doi: 10.1117/12.935640.
[56] R. D. Müller *et al.*, "GPlates: Building a Virtual Earth Through Deep Time," *Geochem. Geophys. Geosyst.*, vol. 19, no. 7, pp. 2243–2261, Jul. 2018, doi: 10/gd7zdc.
[57] A. Tikhonova, C. D. Correa, and K.-L. Ma, "Explorable images for visualizing volume data," in *2010 IEEE Pacific Visualization Symposium (PacificVis)*, Taipei, Taiwan, Mar. 2010, pp. 177–184, doi: 10.1109/PACIFICVIS.2010.5429595.
[58] S. Frey, F. Sadlo, and T. Ertl, "Explorable Volumetric Depth Images from Raycasting," in *2013 XXVI Conference on Graphics, Patterns and Images*, Arequipa, Peru, Aug. 2013, pp. 123–130, doi: 10.1109/SIBGRAPI.2013.26.
[59] A. C. Telea, *Data visualization: principles and practice*. Boca Raton, FL, USA.: AK Peters/CRC Press, 2007.
[60] S. Raman, O. Mishchenko, and R. Crawfis, "Layers for Effective Volume Rendering," in *Proceedings of the Eurographics / IEEE VGTC*

Workshop on Volume Graphics 2008, Los Angeles, California, USA, 2008, p. 7, doi: DOI: 10.2312/VG/VG-PBG08/081-088.
[61] P. E. Morse, A. M. Reading, and T. Stål, "Well-Posed Geoscientific Visualization Through Interactive Color Mapping," *Front. Earth Sci.*, vol. 7, p. 274, Oct. 2019, doi: 10/ggbjzq.
[62] A. Graciano, A. J. Rueda, and F. R. Feito, "Real-time visualization of 3D terrains and subsurface geological structures," *Advances in Engineering Software*, vol. 115, pp. 314–326, Jan. 2018, doi: 10/gf4kr2.
[63] N. Svakhine, D. S. Ebert, and D. Stredney, "Illustration motifs for effective medical volume illustration," *IEEE Computer Graphics and Applications*, vol. 25, no. 3, pp. 31–39, May 2005, doi: 10/fs69s3.
[64] P. Rautek, S. Bruckner, and E. Groller, "Semantic Layers for Illustrative Volume Rendering," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1336–1343, Nov. 2007, doi: 10/cbrwsr.
[65] N. Smit and S. Bruckner, "Towards Advanced Interactive Visualization for Virtual Atlases," in *Biomedical Visualisation*, vol. 1156, P. M. Rea, Ed. Cham: Springer International Publishing, 2019, pp. 85–96.
[66] N. Rawlinson, S. Pozgay, and S. Fishwick, "Seismic tomography: a window into deep Earth," *Physics of the Earth and Planetary Interiors*, vol. 178, no. 3–4, pp. 101–135, 2010, doi: 10/fc6znx.
[67] T. W. Becker and L. Boschi, "A comparison of tomographic and geodynamic mantle models," *Geochemistry, Geophysics, Geosystems*, vol. 3, no. 1, 2002, doi: 10/d7qpv3.
[68] A. M. Dziewonski and J. H. Woodhouse, "Global Images of the Earth's Interior," *Science*, vol. 236, no. 4797, pp. 37–48, Apr. 1987, doi: 10/d4c5x7.
[69] R. Englund and T. Ropinski, "Quantitative and Qualitative Analysis of the Perception of Semi-Transparent Structures in Direct Volume Rendering: Analysis of Perception in DVR," *Computer Graphics Forum*, vol. 37, no. 6, pp. 174–187, Sep. 2018, doi: 10/gdvk6q.
[70] I. Cho, Z. Wartell, W. Dou, X. Wang, and W. Ribarsky, "Stereo and motion cues effect on depth perception of volumetric data," San Francisco, California, USA, Mar. 2014, p. 901118, doi: 10.1117/12.2037942.
[71] W. Hibbard and D. Santek, "Interactivity is the key," in *Proceedings of the 1989 Chapel Hill workshop on Volume visualization - VVS '89*, Chapel Hill, Noth Carolina, United States, 1989, pp. 39–43, doi: 10.1145/329129.329356.
[72] K. Hosseini, K. J. Matthews, K. Sigloch, G. E. Shephard, M. Domeier, and M. Tsekhmistrenko, "SubMachine: Web-Based Tools for Exploring Seismic Tomography and Other Models of Earth's Deep Interior," *Geochemistry, Geophysics, Geosystems*, vol. 19, no. 5, pp. 1464–1483, May 2018, doi: 10/gdt6q5.
[73] G. Zotti, A. Wolf, and F. Chereau, "Stellarium User Guide 0.20.2," 2020. https://github.com/Stellarium/stellarium (accessed Jun. 26, 2020).
[74] T. Butterworth, A. Marini, and S. Contributors, "Syphon Framework," 2018. https://github.com/Syphon/syphon-framework (accessed Jun. 26, 2020).
[75] Adobe Systems Incorporated, *PDF Reference*, 1.7. Adobe Systems Incorporated, 2006.
[76] P. E. Morse, "pemorse/data-visualization-tools: Gradient Designer Suite," Jan. 13, 2020. https://doi.org/10.5281/zenodo.3264036 (accessed Jan. 20, 2020).
[77] T. W. Becker, "SMEAN2 Composite Tomography Model," *Chair in Geophysics: Seismic Tomography Repository*, 2016. http://www-udc.ig.utexas.edu/external/becker/tdata.html#smean2 (accessed Jun. 02, 2020).
[78] J. Ritsema, A. Deuss, H. J. van Heijst, and J. H. Woodhouse, "S40RTS: a degree-40 shear-velocity model for the mantle from new Rayleigh wave dispersion, teleseismic traveltime and normal-mode splitting function measurements: S40RTS," *Geophysical Journal International*, vol. 184, no. 3, pp. 1223–1236, Mar. 2011, doi: 10/fb75gh.
[79] N. A. Simmons, A. M. Forte, L. Boschi, and S. P. Grand, "GyPSuM: A joint tomographic model of mantle density and seismic wave speeds," *Journal of Geophysical Research: Solid Earth*, vol. 115, no. B12, 2010, doi: 10/bkpxvg.
[80] L. Auer, L. Boschi, T. W. Becker, T. Nissen-Meyer, and D. Giardini, "Savani : A variable resolution whole-mantle model of anisotropic shear velocity variations based on multiple data sets," *J. Geophys. Res. Solid Earth*, vol. 119, no. 4, pp. 3006–3034, Apr. 2014, doi: 10/gf2bm6.
[81] M. Gavrilescu and V. Manta, "Advances In The Visualization Of Three-Dimensional Seismic Volume Data.," *Environmental Engineering & Management Journal (EEMJ)*, vol. 10, no. 4, 2011.
[82] T. Stål and A. M. Reading, "A Grid for Multidimensional and Multivariate Spatial Representation and Data Processing," *Journal of Open Research Software*, vol. 8, p. 2, Jan. 2020, doi: 10/ggvr5g.
[83] A. M. Dziewonski and D. L. Anderson, "Preliminary reference Earth model," *Physics of the Earth and Planetary Interiors*, vol. 25, no. 4, pp. 297–356, Jun. 1981, doi: 10/c92pzv.
[84] G. Ranalli, *Rheology of the Earth*, 2nd ed. London, UK: Chapman & Hall, 1995.
[85] D. R. Davies, S. Goes, J. H. Davies, B. S. A. Schuberth, H.-P. Bunge, and J. Ritsema, "Reconciling dynamic and seismic models of Earth's lower mantle: The dominant role of thermal heterogeneity," *Earth and Planetary Science Letters*, vol. 353–354, pp. 253–269, Nov. 2012, doi: 10/gf9vrd.
[86] L. Wang and K. Mueller, "Harmonic Colormaps for Volume Visualization," in *IEEE/ EG Symposium on Volume and Point-Based Graphics*, 2008, p. 7, doi: 10.2312/VG/VG-PBG08/033-039.
[87] L. Wang, J. Giesen, K. T. McDonnell, P. Zolliker, and K. Mueller, "Color Design for Illustrative Visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1739–1754, Nov. 2008, doi: 10.1109/TVCG.2008.118.
[88] R. Montelli, G. Nolet, F. A. Dahlen, and G. Masters, "A catalogue of deep mantle plumes: New results from finite-frequency tomography," *Geochemistry, Geophysics, Geosystems*, vol. 7, no. 11, 2006, doi: 10/cqjmqs.
[89] J. Duncombe, "The Unsolved Mystery of the Earth Blobs," *Eos*, Feb. 2019, doi: 10/gghn3h.
[90] J. Sokol, "Continents of the Underworld Come Into Focus," *Quanta Magazine*, Jan. 2020. https://www.quantamagazine.org/continents-of-the-underworld-come-into-focus-20200107/ (accessed Jun. 02, 2020).
[91] P. Ljung, J. Krüger, E. Groller, M. Hadwiger, C. D. Hansen, and A. Ynnerman, "State of the Art in Transfer Functions for Direct Volume Rendering," *Computer Graphics Forum*, vol. 35, no. 3, pp. 669–691, Jun. 2016, doi: 10/f3r28v.
[92] P. D. Bourke, "Fisheye warping for spherical mirror fulldome projection," *Paul Bourke Personal Pages*, Jul. 2012. http://paulbourke.net/dome/warpingfisheye/ (accessed Jan. 21, 2020).

Peter E. Morse. PhD (1995), BA (Hons) (1987), BA (1984). Previous positions include Lecturer in New Media, Victorian College of the Arts/University of Melbourne (1999-2005); Founding Vice Chair of ACM SIGGRAPH chapters Melbourne (2000-5) and Perth, Australia (2006-7). Antarctic Arts Fellow, Australian Antarctic Division (2005-6). Program Chair ACM SIGGRAPH Graphite 2007, University of Western Australia. Adjunct Senior Research Fellow, iVEC (Pawsey Supercomputing Centre), University of Western Australia (2010-2015). Post-Doctoral Research Fellow, DomeLab, National Institute for Experimental Arts, University of New South Wales (2015). Current: Ph.D. research in scientific data visualization, Compute Earth Group & CODES, Earth Sciences, School of Natural Sciences, University of Tasmania; computer artist and independent consultant specializing in immersive and interactive visualization, Hobart, Tasmania.

Anya M. Reading. Anya M. Reading. PhD (1997), BSc (Hons) (1991). Previous positions include Fellow, Research School of Earth Sciences, Australian National University and Higher Scientific Officer with British Antarctic Survey. Currently employed as Professor of Geophysics, Physics, School of Natural Sciences, University of Tasmania. Fulbright Senior Scholar, 2016. Founder of 'Compute Earth' Group and academic co-lead; Data, Knowledge, Decisions at UTAS. Research focus on computation-enabled understanding of Antarctica, the southern hemisphere continents and the Southern Ocean. Chief Investigator on numerous major research grants pioneering remote area data collection and/or innovative data analysis. Member of numerous learned societies including the Royal Astronomical Society (Geophysics), American Geophysical Union and the Institute of Physics.

Tobias Stål. Tobias Stål, MSc (2015), BSc (2011). Previous work mainly in engineering and applied geophysics, and a career in stage art and development work. Currently PhD student at School of Natural Sciences and Institute for Marine and Antarctic Studies, University of Tasmania. Research interests include integration of geology and geophysics, structural geology, Bayesian methods, open source and collaborative coding.

Chapter 7 : Synthesis and Discussion

7 Overview

The research that underlies this thesis has shown a number of ways in which the visualization of Earth Sciences data can benefit from developments in interactive computer graphics applications, game engines and modern web services. Visualization systems must be able to access the wide variety of data types, file formats and networked database and sensor resources typical of scientific data. These systems must respect underlying data values and enable congruence between those values and visualization display parameters. Visualization displays must take into account aspects of human perception that affect the comprehension of relationships and the recognition of features within data. Interactivity has been shown to exploit and amplify characteristics of perception, such as color relationships, stereopsis and depth perception, as well as temporal processes such as observation of time-based patterns.

This research has been enabled by the development of a suite of interactive computer visualization applications and data pipelines that address the above requirements. The applications were developed specifically for each case study, building interactive interfaces for each type of data under analysis. Through the course of software development and the case studies, the interfaces were iterated and standardized, with a goal of attaining a cross-application consistency in design and interoperability.

Using technological approaches that exploit performant GPU-based graphics, the case studies visualized spatial and temporal features within the three geoscience datasets. Domain expert knowledge, in concert with performant interactive software, yielded new insights into the data. Iterative, controlled interactive visualizations assisted new inferences to be made about the characteristics of features observed. This established the utility of the visualization approaches undertaken and led to new insights into the data being explored and analyzed.

The three core thesis chapters, each published or under review in peer-reviewed journals, describe how these research objectives were met. Each chapter integrates geoscientific data visualization in an analytical workflow and graphics pipeline to

accomplish interpretative and inferential tasks. The software and geoscientific case studies encompassed three levels of dimensionality:

- Chapter 4. The Tagger application demonstrated interactive animated visualization of Waverider Buoy 1D time-series data of the Southern Ocean (Morse et al., 2017)
- Chapter 5. GD and companion apps demonstrated interactive color mapping of 2D depth slices derived from the AuSREM seismic model (Morse et al., 2019)
- Chapter 6. PDT_V and companion apps demonstrated interactive visualization of the volumetric 3D SMEAN2 tomographic dataset of global deep Earth data. (Morse et al., 2020)

The interactive, animated interfaces designed for each type of data visualization built upon developments from previous chapters. Underpinning the applications are shared parallelized dataflow software architectures that were refined and iterated upon throughout the research.

7.1 Relationship to Software

A primary aim of this research was to exploit technologies and modes of HCI from computer graphics and CIT, building upon the feature sets made available by these software environments, and to use them for the discovery of new knowledge in the geosciences. This required extensive evaluation of visualization methodologies and technologies, detailed in Chapters 2 and 3, noting the surprisingly low level of interoperability between approaches and platforms. An important realization was that many CIT applications suitable for developing novel interactive 3D geoscientific visualization interfaces deploy logical, visual and spatial data structures accessible by visual programming architectures (Hughes et al., 2013). This approach is either entirely missing or implemented in limited ways in standard scientific visualization software approaches.

Architectures that articulate a procedural dataflow model facilitate reproduction within a variety of possible development environments, without recommencing from a fundamental programmatic level each time. This has two benefits: it mitigates programming language 'lock in' by abstracting the architecture from the language, and provides a self-documenting template for the architecture of a program. This high-level view can be efficiently reproduced: the template can be used for implementing a program or subroutine in an alternative integrated development environment (IDE) or code-base, such as a computer game engine. However, as this research has demonstrated, these templates also require code-level flexibility, including the development of custom programmed routines for handling and displaying scientific data. This is explored by the hybrid approach utilized in this research.

The applications developed in the course of this research are capable of interacting with

preprocessed data provided in appropriate formats, e.g. ASCII .csv/.txt data hosted locally or as NetCDF on THREDDS servers, or as local or remote raster files created through Python preprocessing (Stål and Reading, 2020). This meets a fundamental requirement of connecting a range of geoscientific data types with the utility of visualization methodologies and workflows. The range of data types can be expanded in future. The capabilities developed for inter-application live graphics sharing, using Syphon (Syphon Developers, 2020), Spout (Spout Developers, 2020) or NDI (NewTek, 2019)) enable the cross-OS manipulation and visualization of data by other types of CIT software, such as XR applications. This programmable graphics-sharing pipeline is a crucial part of the application-level interoperability architecture.

The ability to pipe real-time graphical visualizations of scientific data between computers of varying performance capabilities is a desirable characteristic, that is almost entirely absent from the current scientific visualization toolset. Data preprocessed on high-speed multicore CPUs and graphically processed by high-specification GPUs can be interactively visualized on lower specification systems because those systems only need to interact with streamed video data. This constitutes a performant workflow, whereby data and graphical processing optimizations occur where they can be done most efficiently, with final display and interaction as separable activities for a wide range of potential presentation contexts, including novel modes of display and user-interaction.

Key enablers:

- Research Aim 1: Software has been developed with the ability to load scientific data and display it interactively, exploiting workflows for scientific visualization
- Research Aim 2: Real-time interactive and animated display interfaces have been developed that facilitate 'overview and detail' visualization and navigation of scientific datasets
- Research Aim 5: Developed software suite has the capability of live sharing of visualized data both cross-platform and cross-display applications

7.2 Animated Interactive Display: Time Series

The second aim for this research was to examine ways to amplify analytical acuity through animated and/or interactive interfaces that enable 'overview and detail' reconnaissance and navigation. This parallels the visual analytics 'mantra': "Overview first, zoom and filter, details on demand" (Shneiderman, 1996).

Chapter 4 - *Animated analysis of geoscientific datasets: an interactive graphical application* (Morse et al., 2017) demonstrated that the interactive, animated display of long-run time series data (10 years of Waverider Buoy observations) enabled the analyst to gain new insight into the data that would not have been apparent through a static representation. This was not simply a consequence of whether a static display was used or not, but that the sheer amount of time-based data was impossible to convey in a static image that would fit upon a standard monitor and remain legible, without potentially misleading subsampling.

The presented interactive software, Tagger, enabled the analyst to efficiently load and examine an extensive run of 1D time-series data from a remote THREDDS server into a performant OpenGL GPU-based software tool, and provided the ability to zoom in and out of levels of detail, ranging from a static overview of the entire 10 year dataset (165,484 records, with 2,948,353 data points) down to the level of individual wave events recorded at 30 second intervals capturing 12 parameters per event. Maintaining the <50ms per frame limit for interactivity (Hoetzlein, 2012), GPU-based OpenGL demonstrated two orders of magnitude more capacity than similar data display capacity using HTML5 SVG canvas elements that execute on the CPU (Hoetzlein, 2012; Horak et al., 2018; Morse et al., 2017).

This performant, interactive approach enabled the analyst to discern efficiently features in the data, including new insight into qualitative characteristics of storm events (such as onset and subsidence features), sub-storm events (e.g. periods of elevated activity) and quantitative characteristics such as duration and frequency. Results were found to have a strong association to modelled storm generation scenarios (Hemer, 2010; Hemer et al., 2010), and elicited heretofore unobserved characteristics (e.g. suggesting refined or new types of storm categories) that could be examined subsequently using conventional analysis approach informed by the new insights achieved. Interactive visualization can facilitate cross-matching between modelled scenarios and observed data, elucidating

Key findings include:

- Visualization-enabled reconnaissance of data makes a positive difference in characterization, analysis and knowledge generation
- Animation of time series data facilitates recognition of events and previously unrecognized patterns
- Interactive visualization of time series data with zooming and panning enables greater synoptic overview of long-run data sequences
- Efficient interactive overviews of long runs/large scales of data can be achieved upon desktop computers

commonalities and differences.

7.3 Human in the Loop Visualization

A third aim was to account for 'human in the loop' visualization, considering aspects of human perception in an iterative visualization process. Chapter 5 (Morse et al., 2019) explored the use of perceptual color spaces for optimal display of 2D colorized data. Chapter 6 (Morse et al., 2020) extended this to 3D volumetric visualization, articulating the faculties of stereopsis and depth perception.

Building upon the visualization workflow introduced in Chapter 2, Chapter 5 introduced a concise model of the interactive visualization process, highlighting the iterative process of visualization work in response to both the type of data under consideration and the activity of user-enquiry into that data. The process incorporates important aspects of the human perception of color and how it is used in data visualization. This revealed the need for a methodologically consistent approach to color-mapping that takes into account the non-linear nature of human color perception and an awareness of the problems of color reproducibility, both at a device level and within the context of observation under differing illumination conditions. The research illustrated that a naïve application of RGB color mapping does not linearly correspond to human color perception and runs the risk of misrepresenting data. It introduced the use of the CIELAB perceptually uniform color space (UCS) as an appropriate color space for color-mapping of linear data. Further, it established the utility of the CIELAB color difference metric, ΔE, as an appropriate metric for matching perceived color-difference to underlying data value difference.

A suite of performant, interactive software tools that enabled precise color control within the CIELAB UCS were developed. LAB_CM enabled generation of precise divergent CIELAB gradients. GV_LAB visualized and linearly conformed gradients, within an interactive, animated 3D CIELAB color model display. Accompanying apps were developed for visualizing (non)linear RGB, HSL and HCL trajectories if desirable. GD applied these to 2D data slices from the AuSREM seismic mantle model (Kennett et al., 2013).

These are novel tools for gradient design and optimization for geoscience visualization and for associated workflows that employ inadequate 2D color-picker interfaces. The inherent non-linearity of RGB mapping was demonstrated and quantified. Optimized CIELAB linear color gradients improved upon previous visualization work, providing precisely matching ΔE color correpondence between the underlying data and resultant visualization, including the ability for this metric to represent uncertainty in a consistent, controllable visual way. Finally the use of linearly controlled alpha transparency was introduced, with a view to its use in prospective 3D compositing operations for multiple

layers of data. Developed gradients could be shared live via Syphon (Butterworth et al., 2018), in raster format (.png), data interchange format (JSON) or color palette files (.cpt), widely used by the geoscientific community.

Key findings include:

- Visual inference from color-mapped data must take into account non-linear RGB colorspace and perceptual effects
- CIELAB colorspace provides a close match to human color perception and should be used for linear interpolation between colors representing linear data
- CIELAB colorspace provides repeatable metrics across color reproduction/display devices, including the CIELAB ΔE color difference metric
- Discriminability can be characterized using ΔE
- Uncertainty can be characterized using ΔE, maintaining fidelity to underlying data values

7.4 Well-Posed Methodologies

The visualization developments in animated interactive interfaces (Chapter 4) and precise color control (Chapter 5) laid the foundation for work in Chapter 6 – *Illustrative volumetric deep Earth visualization by 2.5D interactive compositing* (Morse et al., 2020). This introduced the PDT_V 2.5D volumetric visualization application, which interfaced with the other tools developed as part of the GD suite. PDT_V was developed to provide an efficient and intuitive interactive, animated interface for the illustrative visualization of planetary-scale subsurface data. Widely-used applications such as Google Earth and ArcGIS Earth feature no inbuilt capacity to work with such data. It has been, until recently, the domain of complicated and expensive domain-specialist closed-source applications, a variety of discontinued opensource applications, or customized Python-based approaches (Morse et al., 2020). PDT_V radically reimagines such an application approach for future developments in immersive geoscientific visualization environments.

PDT_V was used in a case study to make an illustrative, well-posed visualization of a global geoscience dataset using a combined seismic tomography result, the primary means by which geoscientists infer structure and process in the deep Earth. The computational environment *agrid* (Staal, 2019) was used to generate 58 equirectangular concentric 8-bit raster (.png) slices from the SMEAN2 global, composite mantle tomography model (Becker, 2016). PDT_V visualized these 2.5D slices mapped to appropriately scaled

concentric spheres within the planetary sphere. GD was used to optimize and composite a series of perceptually uniform (LAB_CM) and linear RGB (GD) gradients, monitored for linearity using GV_LAB. This enabled the efficient development of an optimized 2.5D tomographic visualization of deep mantle structure without resort to complicated transfer functions using ray-casting algorithms or 'black box' optical models. The efficient 2.5D compositing operation enabled responsive interactive display upon a mid-range desktop computer, without need of a high-end GPU or large amounts of RAM. The ability to adjust colorization and translucency parameters in real time, in concert with interactive animation of the dataset, articulated powerful depth cues and spatial relationships between subsurface features, including their internal structures, demonstrated in the output animations (Morse, 2020). These features correspond to known entities established in previous studies, generally displayed as monolithic isosurfaces, with newly revealed internal structures. Future development of the interactive display approach using complementary pre-existing seismic tomographic web services such as Submachine (Hosseini, 2019) was discussed.

Key findings include:

- 2.5D compositing can provide controllable color and alpha mapping for 3D tomographic visualization
- 2.5D compositing and alpha control provides effective 3D insight into translucent tomographic seismic models, including embedded structures that may be obscured by isosurface approaches
- Illustrative non-photoreal visualization using complex gradients can highlight significant features and structures in seismic volume data
- Animated 3D/2.5D tomography provides motion cues for powerful depth perception and shape comprehension via human stereopsis

7.5 Well-Posed Visualization

A fourth aim was to develop methodologies and an enriched toolset to facilitate and characterize 'well-posed' geoscientific visualization, including the ability to interface with scientific data formats, to respect underlying data values and to characterize uncertainty.

'Well-posed' scientific visualization arises from the concurrent, managed realization of all these objectives. Software tools were developed that enabled 'overview and detail' review of the data under consideration, maintaining < 50ms per frame draw times for real-time interactivity. This facilitated 'data reconnaissance' whereby the user was able to move efficiently between differing levels of detail to develop an initial overview of the datasets

and zoom to details. Animation of the data (for time series) provided demonstrable benefits for the recognition of patterns, both at a high level (in terms of frequency of events over a long period) and at more granular level in discerning qualitative characteristics of the data. This was similarly evinced with the 2D seismic visualization, where the introduction of perceptually uniform color mapping linearly matched the underlying data values and enabled the visualization of uncertainty using the ΔE metric. This provided the ability to iterate towards a measurably improved visualization of a given dataset, with quantifiable and repeatable color metrics. These two approaches converged in interactive animated visualization software for deep Earth tomographic data. This employed the color-optimized pipeline to accurately visualize SMEAN2 data slices, utilizing animation and three-dimensional depth perception to clearly elicit depth and formal observations of subsurface features of the deep mantle.

Rougier et. al. (2014) provides a clear framework for the implementation of 'well-posed' scientific figures suitable for print or static display, expressed as 10 simple rules. Table 1 adapts these for interactive scientific visualization, drawing on the research presented in this thesis:

	Rules for Well-Posed Visualization	
	Scientific Figures. (Rougier et. al. (2014))	Interactive Visualization
1	Know your audience	Visualization must be sufficiently informational for intended audience (specialist/general)
2	Identify your message	Clear informational purpose of the visualization
3	Adapt the figure to the medium	Design for the platform, including collaboration
4	Captions are not optional	Provide contextual information (e.g. interactive feedback, analogous representations)
5	Do not trust the defaults	Optimize the visualization system for the data and viewer
6	Use color effectively	Awareness of perceptual and cognitive characteristics of color, form and motion.
7	Do not mislead the reader	Explicitness of scale and spatiotemporal relationships
8	Avoid "chart junk"	Avoid visual clutter
9	Message trumps beauty	Distil interfaces and visualizations
10	Get the right tool	Adapt or create the tool for the science

Table 1: Rules for Well-Posed Visualization (after Rougier et al., 2014)

1. A visualization is created with consideration for its intended audience, wherein a specialist audience (e.g. geophysicists) will have common knowledge about what an image represents (e.g. seismic data), but it may require further contextual and explanatory information to be comprehensible to a more general audience (Bond et al., 2007)

2. An awareness of the purpose of a visualization. In a scientific paper, this is clearly to support the scientific argument; however, in interactive visualization of data one of the main purposes is to explore a dataset, to assist observation of features and to inform scientific inference throughout the visualization activity (Sivarajah et al., 2013).
3. An interactive visualization needs to be designed or adapted for the platform or medium upon which it is displayed. These present significant challenges across a range of screen devices and possible modes of interaction. Consideration of collaborative contexts for visualization is an important consideration.
4. Contextual information in an interactive software system provides user feedback, especially those that allow reproducible workflows during interactive enquiry into data. This may take the form of, e.g. logs, annotations or preferences that can be saved, restored and shared.
5. To optimize the visualization for the data requires an awareness of the efficacy of a visualization and the development of a quality criterion (Behrisch et al., 2018). Dimensionality is an important consideration, where, for example, a temporal or spatial dataset might be appropriately displayed via an animated or three-dimensional screen display.
6. Perceptual characteristics of color can have a profound impact upon the interpretation of data (Fairchild, 2013). Understanding the non-linear nature of human color perception is imperative, especially where activities like color-mapping occur (Welland et al., 2006; Ware et al., 2018). Awareness of physiological constraints, like color blindness for certain users, is also an important consideration (Crameri, 2018).
7. Scale and spatiotemporal relationships between forms need to be explicit and clearly displayed in an interactive visualization – especially in 3D interfaces that allow for zooming and panning across data (Shneiderman et al., 2016). Multiple linked views of a data visualization can clarify perceptually confusing relationships (e.g. 3D representations displayed upon 2D screens). Time relationships can take many forms (Aigner et al., 2007) and may be assisted by techniques derived from animation.
8. Visual clutter describes the tendency for visualizations and accompanying software interfaces to become overcomplicated and potentially obfuscating (Few, 2015; Jansen, 1998). Strategies for eliminating these in interactive interfaces and in data strategies like dimensional reduction, filtering and mapping are appropriate.
9. Interfaces and visualizations can be simplified and distilled through an iterative process (Kovesi et al., 2013; Robinson et al., 2018; Wongsuphasawat et al., 2016). Concurrent design refinement of software and good design choices during

development are imperative. This may include abstract dataflow templates that can be reimplemented using varying code-bases and algorithms (Green and Petre, 1996; Mei et al., 2018).

10. Scientific disciplines have specific objectives. Tools that can be adapted or created specifically with a discipline focus are attractive, yet equally, too much specificity can limit their flexibility. For instance, flexible, generic IDEs such as game engines and other media authoring software provide useful models for the development of interactive visualization software for science, if they can be efficiently integrated with scientific data formats and algorithms (Reina et al., 2020).

Key enablers:

- Research Aim 3: Developed software accounts for 'human in the loop visualization' through the use of CIELAB color space, as well as exploiting animation for depth perception and stereopsis
- Research Aim 4: Developed software suite facilitates 'well-posed' visualization through an enriched toolset, by interfacing with scientific data formats, accurately representing scientific data values linearly in uniform color space and maintains fidelity through the ΔE metric, with the capacity to characterize uncertainty

7.6 Collaborative and Immersive Visualization

A final aim of this research was to anticipate and explore how geoscientific visualization software can develop beyond conventional WIMP approaches and interact with shared, immersive and large-scale screen technologies for collaborative work. Two approaches were undertaken: the first for Virtual Reality (VR), developed a simple user-interface for the Tagger software deployed within a VR context using a LEAP motion-controller gestural interface (Morse et al., 2015); the second developed a pipeline for PDT_V to display an appropriately reprojected live video stream for Dome projection (Morse et al., 2020).

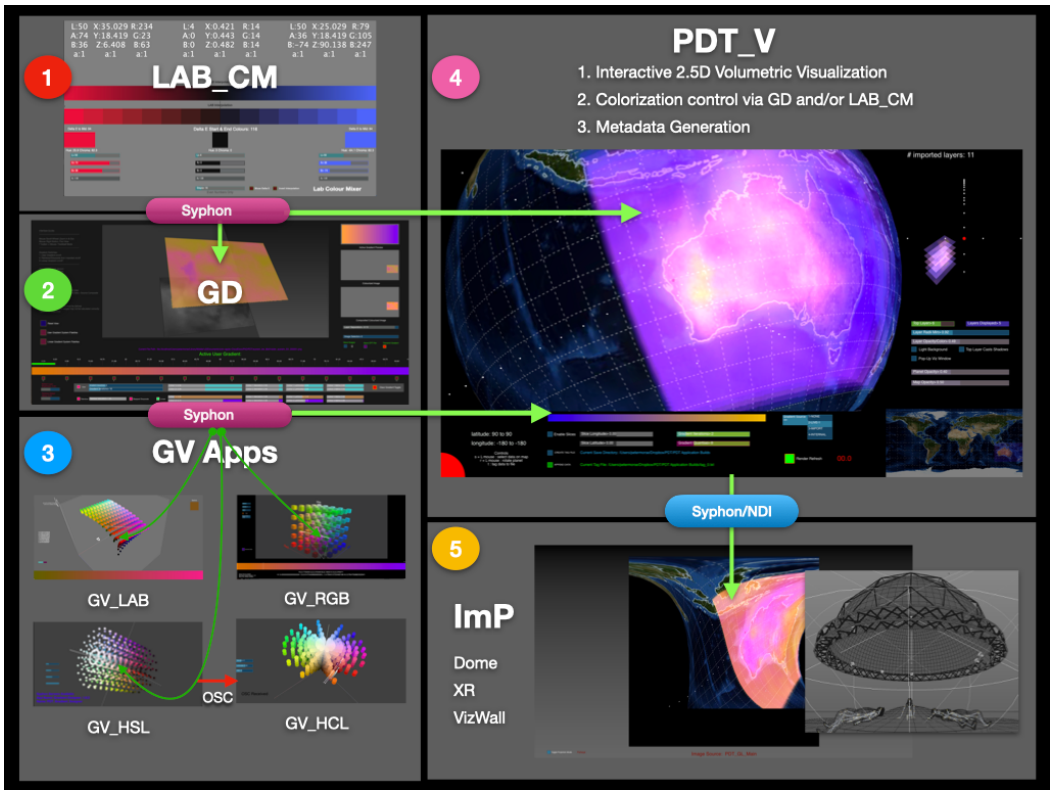


Fig. 1: Application suite workflow: 1) LAB Color Mixer (LAB_CM); 2) Gradient designer (GD); 3) Gradient Visualizer Apps (GV_LAB, GV_RGB, GV_HSL, GV_HCL); 4) Planetary Data Tagger Volumetric (PDT_V); 5) Immersive Player (ImP)

The final software component introduced in this research was the proof-of-concept Immersive Player (ImP) application (Fig.1, Panel 5). This implemented a basic 'live' content video player, capable of remapping an incoming Syphon video stream to both mirrordome and fisheye projection formats (Bourke, 2004). It enables the projection of interactive PDT_V visualizations to large-screen immersive projection systems such as Fulldome (Kwasnitschka, 2017). It demonstrated the capacity to distribute live interactive visualization beyond conventional screen technologies, into large scale collaborative contexts and public outreach situations, such as planetariums. Through the use of the Syphon video streaming capabilities, the application suite can be further developed for XR collaborative interactive environments authored in game engines, including virtual and augmented reality.

Alternative visualization platforms such as XR, CAVE and Dome environments enable data visualization and interrogation in shared collaborative spaces. Research in immersive analytics (IA) platforms (Bach et al., 2017; Marai et al., 2016; Marriott et al., 2018) shows demonstrable benefits for the comprehension of spatial data (Fonnet and Prié, 2019), in particular those displaying depth-relationships, such as geoscience data. Engaging scientists' sensory systems with a capacity for high-dynamic-range color, stereopsis, embodied interaction and haptics suggests ways of augmenting conventional 2D desktop-

based workflows of visualization to assist scientific intuition and inference (Wernert et al., 2012). Immersive, interactive and shared environments will bring more eyes and minds to bear upon current scientific problems. For this to occur, geoscientists require sophisticated software tools, with accountable procedural dataflow architectures, that can adapt efficiently for the constantly evolving hardware and software visualization landscape.

Key enablers:

- Research Aim 5: Developed software suite can present visualized data in collaborative visualization environments such as shared display XR systems and large-scale screen environments such as immersive Domes

7.7 Limitations

The attractive flexibility of a procedural dataflow approach with high-level graphics and interaction control informed the initial choice of development environment under OSX 10.9 (Mavericks) at the commencement of this research. The applications were developed using the MacOS Quartz Composer 4.6.2 (QC) developer environment, which provides an API for MacOS graphics technologies. They use Quartz 2D, OpenGL 3D graphics, and Objective-C processing nodes, as well as some custom routines in GLSL, OpenCL, and JavaScript. As of MacOS 10.15 (Catalina) the QC framework has been deprecated, although it is still present for compatibility. The compiled MacOS applications, code and related datasets are publicly available (Morse, 2018, 2020).

The underlying architecture of the software provides a template for future development, with structural and functional equivalents to common procedural dataflow architectures in CIT IDEs. Current key candidates are the cross-platform procedural visualization software Houdini (SideFX Software, 2020) and/or Touch Designer (Derivative Inc., 2019), and the game engine VPLS, Unity Bolt (Ludiq, 2020) and Unreal Blueprints (Epic Games, 2020a).

7.8 Future Development

Given the MacOS-dependent nature of QC and related graphics APIs, it remains a challenge to determine which platforms will be most suitable for future development, given the advantages and disadvantages of different approaches and their underpinning technologies. Software is a constantly moving target, computer OSes evolve and change, computer languages and functions are deprecated and replaced by new approaches. This

can be ameliorated by software encapsulation in virtual machines or OS-level virtualization (e.g. Docker (Docker Inc., 2020)), delaying the inevitable desuetude. A robust strategy may be found in opensource approaches using game engines, as they can encapsulate all required graphics technologies in an IDE, matched with high-level procedural dataflow and scenegraph models.

Game engines such as Unity (Unity Technologies, 2020), Unreal (Epic Games, 2020) and Godot (Linietsky et al., 2020), have matured since the commencement of this research, to a point where they are now suitable for scientific programming and visualization. This includes support for interactive graphical user interfaces for novel screen technologies such as Dome and XR (Morse et al., 2020) in immersive analytics applications (Morse et al., 2015). This is appealing for future development as it can exploit game engine features (e.g. graphics architectures for interactive 3D GUIs, non-standard display output, GPU shaders) and can mirror the live-code rapid development approach utilized by QC. Future development can leverage their modern graphics APIs and multi-platform support. As discussed in Chapter 3, these developer features are not broadly supported in traditional scientific programming environments such as Python, by scientific libraries and applications such as MatPlotLib, Glue, VisPy, Mayavi, VTK, Paraview, VisIT, Workbench, Matlab and others, nor in modern web technologies such as JavaScript, D3.js, Three.js or WebGL.

This implies that game engine IDEs that employ standardized widely used programming languages such as C++, Python, and OpenGL are prime contenders for mid-term cross-platform development and future support. For an IDE that can compile for OpenGL/Vulkan/Metal graphics and provide game-engine type functionality (including e.g. standardized advanced features for user-interaction, GUI implementation and compute shaders), the dominant options are the closed-source Unity game engine (Unity Technologies, 2020), source-available C++ -based Unreal Engine (Epic Games, 2020), and the Python-compliant opensource Godot engine (Linietsky et al., 2020). These afford the most tractable options for multiplatform, multidevice support (e.g. MacOS, iOS, Windows, Linux, Android) with a credible roadmap and critical-mass developer community for ongoing opensource development. Importantly, they now feature varying levels of support for interfacing with Python, the key programming language for many scientific software libraries used in the geosciences.

Building upon recent practical experience, future software development anticipates use of the source-available Unreal Engine (Epic Games Inc., 2020). Over recent years, this has developed into a technically advanced high-performance visualization engine, with a superior system-wide VPL, and cross-platform compatibility. The graphics and programming feature set includes modern APIs and advanced shading language support, including a native Python API that can be directly interfaced with scientific workflows.

The visualization opportunities opened by the exploratory approaches we have demonstrated, taking into account complex features of human perception and human-computer interaction, suggest that many pathways towards novel forms of scientific visualization remain to be developed in detail. Affordances for collaborative scientific workflows can be developed by exploiting the largely untapped resources of game engine environments and advanced GPU processing, potentially leading to significant new scientific insights as a consequence of advanced scientific visualization and visual analytics workflows. The reconnaissance and exploration of geoscientific data through these innovative approaches, where visualization forms a significant component of inference, is an attractive and exciting opportunity for future research.

7.9 References

Aigner, W., Miksch, S., Müller, W., Schumann, H., Tominski, C., 2007. Visualizing time-oriented data—A systematic view. Computers & Graphics 31, 401–409. https://doi.org/10.1016/j.cag.2007.01.030

Bach, B., Cordeil, M., Dwyer, T., Lee, B., Saket, B., Endert, A., Collins, C., Carpendale, S., 2017. Immersive analytics: Exploring future visualization and interaction technologies for data analytics, in: IEEE VIS, Accepted Workshop. https://doi.org/10.1145/2992154.2996365

Becker, T.W., 2016. SMEAN2 Composite Tomography Model [WWW Document]. Chair in Geophysics: Seismic Tomography Repository. URL http://www-udc.ig.utexas.edu/external/becker/tdata.html#smean2 (accessed 6.2.20).

Behrisch, M., Blumenschein, M., Kim, N.W., Shao, L., El-Assady, M., Fuchs, J., Seebacher, D., Diehl, A., Brandes, U., Pfister, H., Schreck, T., Weiskopf, D., Keim, D.A., 2018. Quality Metrics for Information Visualization. Computer Graphics Forum 37, 625–662. https://doi.org/10.1111/cgf.13446

Bond, C.E., Gibbs, A.D., Shipton, Z.K., Jones, S., 2007. What do you think this is? "Conceptual uncertainty" in geoscience interpretation. Gsa Today 17, 4. https://doi.org/10/d9nvnx

Bourke, P., 2004. Dome projection on a budget (MirrorDome) [WWW Document]. URL http://paulbourke.net/dome/mirrordome/ (accessed 1.27.20).

Butterworth, T., Marini, A., Contributors, S., 2018. Syphon Framework [WWW Document]. URL https://github.com/Syphon/syphon-framework (accessed 6.26.20).

Crameri, F., 2018. Scientific colour maps. Zenodo, Oslo, Norway. https://doi.org/10.5281/zenodo.1243862

Derivative Inc., 2019. Derivative TouchDesigner [WWW Document]. URL https://www.derivative.ca/ (accessed 1.21.20).

Docker Inc., 2020. Empowering App Development for Developers | Docker [WWW Document]. Docker. URL https://www.docker.com/ (accessed 2.11.20).

Epic Games, 2020. Unreal Engine [WWW Document]. Unreal Engine. URL http://www.unrealengine.com/ (accessed 1.21.20).

Epic Games, 2020a. Introduction to Blueprints [WWW Document]. Unreal Engine Blueprints. URL https://docs.unrealengine.com/en-US/Engine/Blueprints/GettingStarted/index.html (accessed 2.11.20).

Fairchild, M.D., 2013. Color Appearance Models Third Edition. John Wiley & Sons.

Few, S., 2015. Signal: Understanding what Matters in a World of Noise. Analytics Press.

Fonnet, A., Prié, Y., 2019. Survey of Immersive Analytics. IEEE Transactions on Visualization and Computer Graphics 1–1. https://doi.org/10/gf73gm

Green, T.R.G., Petre, M., 1996. Usability Analysis of Visual Programming Environments: A 'Cognitive Dimensions' Framework. Journal of Visual Languages & Computing 7, 131–174. https://doi.org/10.1006/jvlc.1996.0009

Hemer, M.A., 2010. Historical trends in Southern Ocean storminess: Long-term variability of extreme wave heights at Cape Sorell, Tasmania: SOUTHERN OCEAN STORMINESS TRENDS. Geophysical Research Letters 37, n/a-n/a. https://doi.org/10.1029/2010GL044595

Hemer, M.A., Church, J.A., Hunter, J.R., 2010. Variability and trends in the directional wave climate of the Southern Hemisphere. International Journal of Climatology 30, 475–491. https://doi.org/10/bqqxdb

Hoetzlein, R.C., 2012. Graphics Performance in Rich Internet Applications. IEEE Computer Graphics and Applications 32, 98–104. https://doi.org/10.1109/MCG.2012.102

Horak, T., Kister, U., Dachselt, R., 2018. Comparing Rendering Performance of Common Web Technologies for Large Graphs, in: Poster Program of the 2018 IEEE VIS Conference, VIS. p. 2.

Hosseini, K., 2019. SubMachine: Web-based tools for exploring seismic tomography and other models of Earth's deep interior [WWW Document]. SubMachine: Web-based tools for exploring seismic tomography and other models of Earth's deep interior. URL https://doi.org/10.1029/2018GC007431 (accessed 7.4.19).

Hughes, J.F., Van Dam, A., Foley, J.D., McGuire, M., Feiner, S.K., Sklar, D.F., Akeley, K., 2013. Computer graphics: principles and practice. Addison-Wesley Professional.

Jansen, B.J., 1998. The Graphical User Interface. SIGCHI Bull. 30, 22–26. https://doi.org/10/c25jxj

Kennett, B.L.N., Fichtner, A., Fishwick, S., Yoshizawa, K., 2013. Australian Seismological Reference Model (AuSREM): mantle component. Geophysical Journal International 192, 871–887. https://doi.org/10.1093/gji/ggs065

Kosada Inc., 2019. Vuo [WWW Document]. Vuo. URL https://vuo.org/ (accessed 1.21.20).

Kovesi, P., Holden, E.-J., Wong, J.C., 2013. Interactive multi-image blending for data visualisation and interpretation. ASEG Extended Abstracts 2013, 1–4. https://doi.org/10/ghz5ct

Kwasnitschka, T., 2017. Planetariums — not just for kids. Nature 544, 395–395. https://doi.org/10/gf4ksh

Linietsky, J., Manzur, A., Contributors, 2020. Godot Engine - Free and open source 2D and 3D game engine [WWW Document]. URL https://godotengine.org/ (accessed 1.21.20).

Ludiq, 2020. Bolt: Visual Scripting for Unity [WWW Document]. Ludiq. URL https://ludiq.io/bolt (accessed 2.11.20).

Marai, G.E., Forbes, A.G., Johnson, A., 2016. Interdisciplinary immersive analytics at the electronic visualization laboratory: Lessons learned and upcoming challenges, in: 2016 Workshop on Immersive Analytics (IA). Presented at the 2016 Workshop on Immersive Analytics (IA), IEEE, Greenville, SC, USA, pp. 54–59. https://doi.org/10.1109/IMMERSIVE.2016.7932384

Marriott, K., Chen, J., Hlawatsch, M., Itoh, T., Nacenta, M.A., Reina, G., Stuerzlinger, W., 2018. Immersive Analytics: Time to Reconsider the Value of 3D for Information Visualisation, in: Marriott, K., Schreiber, F., Dwyer, T., Klein, K., Riche, N.H., Itoh, T., Stuerzlinger, W., Thomas, B.H. (Eds.), Immersive Analytics, Lecture Notes in Computer Science (LNCS. Springer International Publishing, pp. 25–55. https://doi.org/10.1007/978-3-030-01388-2_2

Mei, H., Chen, W., Ma, Y., Guan, H., Hu, W., 2018. VisComposer: A Visual Programmable Composition Environment for Information Visualization. Visual Informatics, Proceedings of PacificVAST 2018 2, 71–81. https://doi.org/10/gf4kmj

Morse, P., 2018. pemorse/Tagger [WWW Document]. URL https://github.com/pemorse/Tagger (accessed 2.11.20).

Morse, P., Reading, A., Lueg, C., 2017. Animated analysis of geoscientific datasets: An interactive graphical application. Computers & Geosciences 109, 87–94. https://doi.org/10/gcmxvw

Morse, P., Reading, A., Lueg, C., Kenderdine, S., 2015. TaggerVR: Interactive Data Analytics for Geoscience - A Novel Interface for Interactive Visual Analytics of Large Geoscientific Datasets in Cloud Repositories, in: 2015 Big Data Visual Analytics (BDVA). Presented at the 2015 Big Data Visual Analytics (BDVA), IEEE, Hobart, Australia, pp. 1–2. https://doi.org/10.1109/BDVA.2015.7314303

Morse, P.E., 2020. pemorse/data-visualization-tools: Gradient Designer Suite [WWW Document]. https://doi.org/10.5281/zenodo.3264037

Morse, P.E., Reading, A.M., Stål, T., 2020. Under Review: Illustrative volumetric deep Earth visualization by 2.5D interactive compositing. IEEE Transactions on Visualization and Computer Graphics 14 pages.

Morse, P.E., Reading, A.M., Stål, T., 2019. Well-Posed Geoscientific Visualization Through Interactive Color Mapping. Front. Earth Sci. 7, 274. https://doi.org/10/ggbjzq

NewTek, 2019. NewTek NDI Tools [WWW Document]. NewTek. URL https://www.newtek.com/ndi/tools/ (accessed 2.22.19).

Reina, G., Childs, H., Matković, K., Bühler, K., Waldner, M., Pugmire, D., Kozlíková, B., Ropinski, T., Ljung, P., Itoh, T., Gröller, E., Krone, M., 2020. The moving target of visualization

software for an increasingly complex world. Computers & Graphics 87, 12–29. https://doi.org/10/ggk8w4

Robinson, J., Lanius, C., Weber, R., 2018. The past, present, and future of UX empirical research. Communication Design Quarterly Review 5, 10–23. https://doi.org/10.1145/3188173.3188175

Rougier, N.P., Droettboom, M., Bourne, P.E., 2014. Ten Simple Rules for Better Figures. PLOS Computational Biology 10, e1003833. https://doi.org/10.1371/journal.pcbi.1003833

Shneiderman, B., 1996. The eyes have it: a task by data type taxonomy for information visualizations, in: Proceedings 1996 IEEE Symposium on Visual Languages. Presented at the Proceedings 1996 IEEE Symposium on Visual Languages, pp. 336–343. https://doi.org/10/fwdq26

Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., Diakopoulos, N., 2016. Designing the User Interface: Strategies for Effective Human-Computer Interaction, 6th ed. Pearson.

SideFX Software, 2020. Houdini - 3D modeling, animation, VFX, look development, lighting and rendering | SideFX [WWW Document]. URL https://www.sidefx.com/ (accessed 2.11.20).

Sivarajah, Y., Holden, E.-J., Togneri, R., Dentith, M., 2013. Identifying effective interpretation methods for magnetic data by profiling and analyzing human data interactions. Interpretation 1, T45–T55. https://doi.org/10/ghz5bh

Spout Developers, 2020. Spout [WWW Document]. Spout. URL https://spout.zeal.co/ (accessed 2.11.20).

Staal, T., 2019. TobbeTripitaka/*agrid* [WWW Document]. GitHub. URL https://github.com/TobbeTripitaka/agrid (accessed 1.21.20).

Stål, T., Reading, A.M., 2020. A Grid for Multidimensional and Multivariate Spatial Representation and Data Processing. Journal of Open Research Software 8, 2. https://doi.org/10/ggvr5g

Syphon Developers, 2020. Syphon is a Mac OS X technology to allow applications to share video and still images with one another in realtime, instantly.: Syphon/Syphon-Framework [WWW Document]. URL https://github.com/Syphon/Syphon-Framework (accessed 2.11.20).

Unity Technologies, 2019. Unity [WWW Document]. Unity. URL https://unity.com/frontpage (accessed 1.21.20).

Ware, C., Turton, T.L., Bujack, R., Samsel, F., Shrivastava, P., Rogers, D.H., 2018. Measuring and Modeling the Feature Detection Threshold Functions of Colormaps. IEEE Transactions on Visualization and Computer Graphics 25, 2777–2790. https://doi.org/10.1109/TVCG.2018.2855742

Welland, M., Donnelly, N., Menneer, T., 2006. Are we properly using our brains in seismic interpretation? The Leading Edge 25, 142–144. https://doi.org/10/b3skdx

Wernert, E.A., Sherman, W.R., O'leary, P., Whiting, E., 2012. A Common Path Forward for the Immersive Visualization Community, in: Proceedings of IEEE VR 2012. Presented at the Workshop Immersive Visualization Revisited Challenges and Opportunities at IEEE VR 2012, Costa Mesa, California, USA, pp. 1–8.

Wongsuphasawat, K., Moritz, D., Anand, A., Mackinlay, J., Howe, B., Heer, J., 2016. Voyager: Exploratory Analysis via Faceted Browsing of Visualization Recommendations. IEEE Transactions on Visualization and Computer Graphics 22, 649–658. https://doi.org/10.1109/TVCG.2015.2467191

Chapter 8: Conclusions

The research presented in this thesis has progressed the practical application of interactive visualization to the scientific analysis of geoscience data. It proposed a model of the relationships between data, user and visualization, and used this to characterize a workflow for the visualization process. This workflow informed the technical implementation of a pipeline for interactive data visualization. The pipeline defined a technical brief for hardware choices and software development, resulting in applications that realized novel visualization tools and methodologies. A notable aspect of the conceptual intent of the research, and its technical implementation, was the connection of scientific data inputs and visualization requirements to the flexible software prototyping capabilities enabled by approaches from creative industries technologies. This enabled exploration of the interactive visualization design space, customizable workflows, reconfigurable display formats and user-interfaces. This human-centered computing approach leveraged important aspects of human perception and cognition. These include motion awareness, perceptually-uniform color spaces, stereopsis and depth perception.

A suite of new software applications was developed that employ animation, realtime interaction, and precise color control. They include the ability to capture relevant metadata through human interaction and feature observation annotations. The software includes innovative features such as the ability to live-share visualization parameters, colormaps and animations to external client applications. These principally focused on a standard display with a performant desktop computer, to be readily accessible to individuals or groups carrying out geoscience research. Application functionality, however, also extends to outputs compatible with collaborative and immersive graphics environments. Output for multi-screen visualization walls, dome displays and XR clients enable application utilization for collaborative data inference as well as the engaging display of geospatial research results.

Case studies in the three core research chapters demonstrated the analysis of time-variant ocean storm data, 2D spatial data in the form of slices through a seismic wavespeed model for Australia, and 3D volumetric data in the form of global seismic tomography models.

The first case study applied data-driven visual augmentation and time-based data animation, of a large time-series of ocean storm data. In concert with overview and detail

reconnaissance, this approach enabled the characterization of newly recognized ocean storm features.

The second case study used interactive color-mapping applied to seismic wavespeed depth-slices. This was performed in perceptual CIELAB color-space, utilizing ΔE for the creation of perceptually uniform gradients. This enabled the development of linear, quantifiable and perceptually accurate color-maps for the well-posed visualization of features within the data representation, improving upon the representations used in prior work. The application suite permits the precise visualization of color-map trajectories through multiple color-space representations, assisting optimization of color-mapping for human visual perception in different display contexts. Color-map output for geoscience standards such as GMT color palette format ensure interoperability with geoscientific workflows and assists reproducibility. In doing so it encourages attention to both "visual literacy" and "visual numeracy" for geoscientific data visualization.

The third core research chapter applied the well-posed use of color, to a global seismic tomography result, visualizing the deep mantle structure of the Earth. The intuitive 2.5D layer approach employed, in concert with realtime interactive animation of the Earth sphere, assisted depth-perception and spatial characterization of feature relationships within the dataset. It contrasted numerical and perceptual color optimization approaches to the compositing of layered translucent depth-slices for volumetric visualization. It employed a human-in-the-loop approach to reconnaissance and interactive exploration of translucent embedded tomographic structures. Interaction with online web-services such as Submachine were proposed, and prototype applications for dome and XR were demonstrated in an accompanying video. This illustrates the complete pipeline detailed in Chapter 3, from data, through well-posed interactive desktop visualization, to immersive screen display.

In all case studies the software developed included basic annotation capabilities in the form of user-definable tags written to simple text files. For time-based data they included relevant time stamp and selection ranges for temporal feature identification, to spatial data they included relevant spherical coordinate information. Appropriate and flexible metadata generation during data reconnaissance can connect visual exploration activity back to conventional analyses as required.

The research undertaken in this thesis has demonstrated significant achievements in progressing the utility of interactive scientific visualization for the geosciences. These include:

- Demonstration of improved analytical acuity through animation, overview and detail visualization (Chapter 4)
- Development of methods for well-posed visualization, including a linear metric, ΔE, for color-difference and uncertainty representation (Chapter 5)
- Accounting for human-in-the-loop visualization by optimizing for human color perception, stereopsis and depth perception (Chapters 5 and 6)

- Precise control of visual representation of underlying data sources, including the ability to demonstrate non-linearities in various color-spaces and compositing operations (Chapters 5 and 6)
- Ability to compensate visually for non-linear compositing operations (Chapter 6)
- Ability to define rigorously color-maps using CIELAB values for machine-independent reproducibility
- Ability to interface with scientific data formats through network and local delivery of data, methodologies for preprocessing and format conversion, if necessary
- Ability to write-out open-format geoscientific file types
- Ability to generate open-format metadata for conventional analytical approaches
- Adaptable UI development approaches for collaborative, immersive and large-scale screen technologies

The underpinning VPL architecture developed in this research facilitates reproduction within a variety of possible development environments. By preserving high-level abstractions of software, a VPL approach assists the rapid transfer of algorithms in the fast-evolving CIT hardware and software ecosystem. This has two benefits: it mitigates programming language 'lock in' by abstracting the architecture from the language, and provides a self-documenting template for the architecture of a program.

The practical convergence of interactive scientific visualization with the cutting-edge capabilities of modern CIT approaches heralds an exciting future for geoscientific data analysis. Immersive, interactive and shared environments will bring more eyes and minds to bear upon current scientific problems. For this to occur, geoscientists require sophisticated software tools, with accountable procedural dataflow architectures, that can adapt efficiently for the constantly evolving hardware and software visualization landscape. Interactive visualization systems, rigorously informed by the perceptual, cognitive and technical aspects considered in this research, have been shown to improve the process of scientific inference. This will enrich the understanding of datasets representing aspects of the natural, physical world, including the Earth's oceans and deep planetary interior.

Supplementary Material

This section includes supplemental print material referenced by the relevant chapters, below:

Supplement 1: Chapter 4. Supplementary Material

Supplement 2: Chapter 6. Supplementary Material

Supplement 3: A brief technical appendix.

Online Supplementary materials, including software, movies and high-definition images, are available at:

https://github.com/pemorse/data-visualization-tools

and

https://doi.org/10.5281/zenodo.3264037

Supplementary Material: Chapter 4

Animated Analysis of Geoscientific Datasets:

An Interactive Graphical Application

Peter Morse[1*], Anya Reading[1], Christopher Lueg[2]

[1] School of Physical Sciences (Earth Sciences), University of Tasmania, Hobart, TAS, 7001, Australia
[2] School of Engineering and ICT, University of Tasmania, Hobart, TAS, 7001, Australia

Supplements

1 Supplement 1: Technical Background

1.1 GPU-accelerated graphics

OpenGL is the industry standard for real-time GPU-accelerated graphics rendering (Neider, Davis & Woo 1993). It provides a software interface to graphics hardware, using an open specification that is supported by every major operating system and the three major hardware vendors (NVIDIA, Intel, AMD). OpenGL defines the graphics pipeline by which 2D and 3D data are processed on the GPU, before being drawn to screen, based upon fundamental specifications of geometric primitives, such as points, lines and polygons. In addition to geometry it can also specify depth-cues, antialiasing, shading models (including color, lighting and shadows), textures, motion blur and many other visual features, finally rasterizing these for screen display. Built on top of OpenGL are languages, libraries and APIs, e.g. Open Inventor (Wernecke 1994), GLSL (Kessenich, Baldwin & Rost 2004) and the Visualisation ToolKit VTK (Schroeder, Avila & Hoffman 2000; Kitware 2016). These provide higher-level programmatic abstractions to the underlying OpenGL primitives, and facilitate the implementation of visualisation processes and methodologies by graphics developers. The new open standard Vulkan API (Khronos Group 2015) (previously referred to as the Next Generation OpenGL initiative) is a ground-up redesign of the graphics API for the architecture of modern GPUs. Vulkan is a unified specification for cross-platform high-performance graphics that will complement OpenGL and OpenGL ES.

In parallel with the development of OpenGL during and since the 1990s has been the emergence of ‘Game Engines’ (Cowan & Kapralos 2014) and programming environments for computer graphics (e.g. Quartz Composer ((Apple Computer 2007), Processing (Processing Developers 2016), Cinder (Cinder Developers 2016)). Over the last 20 years as the video game industry and market has matured, game engines and graphics programming environments have co- evolved

closed-source proprietary systems as well as opensource systems supported by significant developer communities (Kosada Inc. 2016; Meloni 2015; Rijnieks 2013; Friese, Herrlich & Wolter 2008).

1.2 Technology choice for interactive graphics

Interactive data visualization has been enabled by open standards such as HTML5, WebGL (Khronos Group 2016) and Javascript (including libraries such as Three.js (ThreeJS Developers 2016), D3.js (Bostock, Ogievetsky & Heer 2011)) and commercial API/data visualisation services (e.g. Google Charts (Google Inc. 2016) and Plotly (Plotly Inc 2016)) via the HTML5 DOM (the 'document object model' parsed by web browsers), leading to an explosion of near-real-time online data visualisation possibilities. A significant constraint in the HTML5 DOM is the use of the Canvas element and SVG, which restricts graphics to a 2D CPU-based render context, placing limitations on the complexity of scenes that can be drawn and animated (Hoetzlein 2012). In developing 'Tagger' we therefore opted to use OpenGL for its performant capacities.

An emerging technology, WebGL (a 3D subset of OpenGL based on OpenGL ES 2.0), although exposed through the Canvas element, is hardware accelerated on the GPU. Comparative benchmarks demonstrate it is far more capable of drawing animated complex scenes requiring millions to billions of points (Halic, Ahn & De 2015). WebGL is now supported by most browsers, operating systems and hardware, and demonstrates significant future promise as the technology matures, particularly via integration with game engines. Although WebGL and Javascript enable 3D, animation and interaction capabilities in the browser, they do not solve the problem of how data are delivered to the client (see: 1.5).

1.3 GPU Accelerated Graphics

Dynamic data representations demand graphics hardware of appropriate specification. Typically, computers need to be capable of screen-refresh rates of at least 20 frames per second (20Hz) or more in order for a user to perceive smooth animation. Animation standards in the computer graphics industry mandate screen-refresh rates of 30-60Hz for monoscopic and 120Hz+ for stereoscopic presentations. Coupled with high-resolution screens, e.g. WQHD (2560x1440px) or 4K (4096x2160), many millions of pixels need to be drawn to screen every second. GPUs are far more capable than CPUs for tasks such as graphics and analogous algorithmic processes that benefit from parallel processing (Gregg & Hazelwood 2011). The development of the GPU

(McClanahan 2010) has been well resourced and driven by the computer graphics, games and animation industries; key technologies are summarised in this Supplement.

1.4 Data Formats for Scientific Research

Data are diverse collections of numbers and characters that represent qualitative and quantitative human observations, measurements made using devices in the field or as part of an experiment, and the results of different kinds of simulations. For these data to be usable, they must be in a readable form and susceptible to programmatic analysis. In general, data formats have been tailored to specific user communities and scientific data formats are often a relic of the hardware specifications of a former era. A recent requirement is that the data may be usefully held and managed in a repository accessible as part of an interoperable database (Gray et al. 2005) to facilitate new approaches to large datasets and computational analysis. General data users and the scientific community are therefore moving away from 'flat' file systems stored on local hard drives (e.g. .csv) which are insufficiently agile and robust for many types of analytical enquiry, particularly relating to very large multidimensional and heterogeneous datasets.

Formats such as NetCDF (Rew & Davis 1990) and HDF5 (Folk et al. 2011) encode data as self-describing matrices, hyperslab and vector addressable high dimensional structures. These interoperable machine-independent abstractions enable the containment and structuring of heterogeneous data and maintain a wide variety of query vectors. Such formats have evolved together with data access methods such as the NCAR Command Language (NCL) (Brown et al. 2012) and NcML mark-up language (Nativi, Caron & Domenico 2004). Together, agile data formats and flexible query languages facilitate analyzing and understanding data in new and insightful ways.

1.5 Network Architecture, Data Servers and Cloud Computing

Network and server architectures are still one of the key bottlenecks in the transfer of data (Hansen & Johnson 2011, pp.512–525). Visualisation of large datasets is intrinsically constrained by bandwidth: compression and down-sampling are frequently necessary if data are delivered over networks (Ahrens et al. 2009). Alternatives include moving the compute to the data (e.g. via cloud computing) or implementing subsampling via protocols such as OPeNDAP.

We make use of a THREDDS Data Server (TDS) for the storage and delivery of data. A TDS (Unidata 2016) is an opensource data architecture developed for the geosciences. THREDDS (Thematic Real-time Environmental Distributed Data Services) are middleware, whose services, provided by the TDS, use a high-level data abstraction, the Common Data Model, enabling unified access to netCDF and HDF5 data models (amongst others) via a common API, implemented in Java (Davis & Caron 2002). OPeNDAP ('Opensource Project for a Network Data Access Protocol') is a data transport architecture and protocol, again widely used in the geosciences. The TDS can interpret OPeNDAP queries (programmatically conformed URLs passed to the server, parsed as queries) in order to request actions of the server upon its catalogue of data (an XML metadata repository). Actions may include retrieving subsets of given data (e.g. a date range, with a series of variables and parameters), virtual concatenations or other aggregations of datasets using NcML, NCL commands and a variety of NetCDF/HDF5 queries, such as slicing and dicing datacubes, and requesting vectors through data. THREDDS and OPeNDAP enable self-describing multidimensional data subsets to be accessed via shared compute resources. This has the advantage of by-passing network bottlenecks and minimizing the need to download the large datasets typical of geosciences.

A feature of large-scale data management, access and analysis is interoperability with 'cloud' computing and data storage (Fox et al. 2009; Mell & Grance 2011). This has been driven by the relatively poor rate of network bandwidth increase versus the rate at which data are being collected (Hey et al. 2009; Emmott & Rison 2006). It may be more efficient to 'move the compute to the data than the data to the compute', in the form of virtual machines (VMs) located in immediate proximity to data repositories, or between machines connected via a high-bandwidth data bus (e.g. Infiniband (Pfister 2001), Fiber Channel (Cummings 1993)). Virtual Machines may be hosted on an IaaS system ('Infrastructure as a service', such as the Amazon AWS).

2 Supplement 2: Tagger -THREDDS Interaction

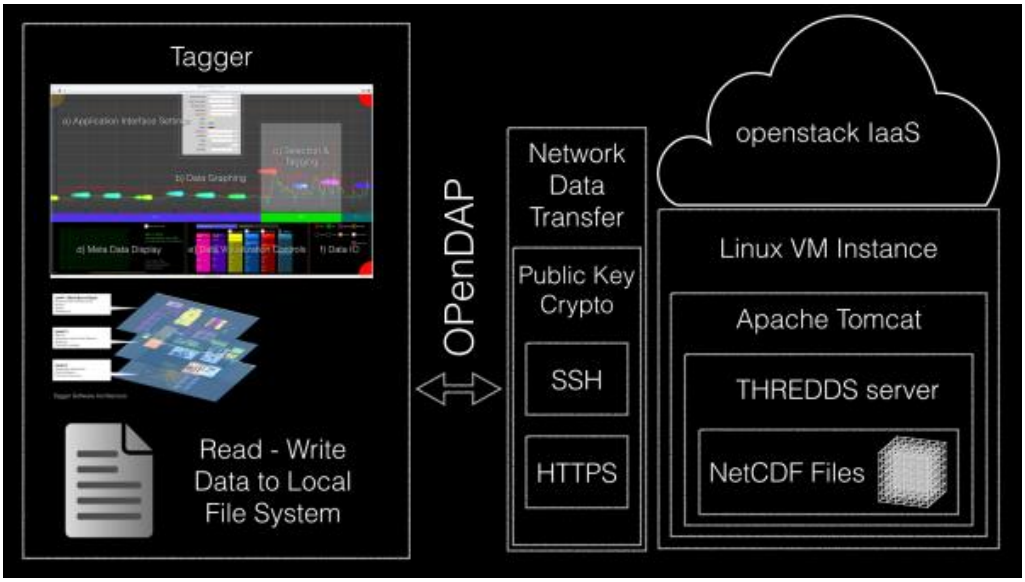


Figure 1: Tagger System Overview

'Tagger' is interoperable with a THREDDS server and NetCDF files, via OPeNDAP, as well as local files (including .txt and .csv formats) (Morse et al. 2015). For this work an Ubuntu Linux VM was deployed on the NCRIS NeCTAR cloud platform (National Research Infrastructure for Australia 2016) and prepared with the appropriate Java runtime environment for hosting a THREDDS data server (TDS), which runs as a servlet in a java-based Apache Tomcat webserver. These NetCDF files were then transferred to the TDS and registered in the THREDDS catalogue using the appropriate XML markup. An OPeNDAP parser was implemented in Tagger that enables properly conformed OPeNDAP requests to be made across a secure network to the TDS. The TDS returns an ASCII datastream over HTTP/HTTPS, in response. Tagger parses the initial header information in the datastream, and buffers the stream in anticipation of user-interaction for graphing and tagging. Buffers can be periodically cleared via user interaction, to avoid memory overflow problems. At this point an end-user can begin the process of observing the data flow and begin to tag, as desired. These tag files are written out to the local file system. Tagger can therefore be used to tag arbitrarily large datasets.

3 Supplement 3: Quartz Composer Overview

Quartz Composer (QC) (Apple Computer 2007) is a Visual Programming Language (VPL) delivered as part of Apple's freely available Xcode development environment (Apple Computer 2016). It was first introduced in Mac OSX 10.4 and remains current, despite slow development since version 4.6. QC provides an interface to underlying technologies such as OpenGL, OpenCL, Core Image, Core Video and other OSX graphics technologies. It is extensible via a plug-in architecture and also permits the development of text-coded routines in Objective-C, OpenCL or Javascript, in concert with the visual programming paradigm. Quartz Compositions (data flow programs) can be distributed as freely-modifiable programs or can be compiled into stand-alone software applications via Xcode. It has recently been extensively used by GUI designers working at Facebook, for the development of Android and iOS application prototypes such as Facebook Home, using their in-house Origami Toolkit (Facebook Design 2016). The VPL paradigm enables a fast-turn-around and easily comprehensible programming style that can be used to iterate design patterns, leading to rapid development and iteration of software prototypes. It makes software customisation available to a wide range of users.

VPL dataflow metaphors (Sutherland 1966; Myers 1990; Sanner, Stoffler & Olson 2002; Sousa 2012) are used in many current dataflow programming environments, including scientific ones such as Workspace (CSIRO 2016), Orange (Demsar et al. 2004; University of Ljubljana 2016), Paraview (Kitware Inc. 2016), OpenInsightExplorer (Stehno 2012), OpenDX (Abram 2000) and many others. VPLs are notable for ease-of-use, logical transparency and rapid prototyping applications. Quartz Composer can be extended via plugins, patches and composition subroutines developed via an extensive user community – most notably Kineme (Kosada Inc. 2016), who provide the ChartTools plugins, extensively used in Tagger. ChartTools provide a QC interface to a private undocumented library in OSX called GraphKit.framework, used in various OS graphing applications, such as Activity Monitor. Given the undocumented nature of this library, for this research alternatives were explored such as the CorePlot libraries, which also include a QC Plugin that can be compiled using XCode (Core-Plot Developers 2016).

Quartz compositions are procedural motion graphics programs created by assembling predefined patches available via a patch library or 'composition repository'. The QC interface uses a cable-patching metaphor to establish data, parameter and control flow between different 'patches' (nodes) in a composition. Patches are similar to routines in traditional programming languages and form the base processing units for a program. They have input and output ports which pass parameters from node to node via interconnecting cables, much like a circuit diagram. Patches

fall broadly into three categories according to their execution mode: *consumer*, *processor* or *provider*. *Consumers* render a result to a destination (e.g. the screen); *processors* perform some operation upon data passed into them (e.g. a mathematical operation); *providers* supply data from an outside source to a composition (e.g. read a text file.) A Quartz composition is similar to any complex C or Objective-C program that has a main routine and many subroutines. Similarly, routines and subroutines can be nested into 'Macro Patches', forming complex patch hierarchies (Apple Computer 2007).

4 Supplement 4: Tagger Architecture

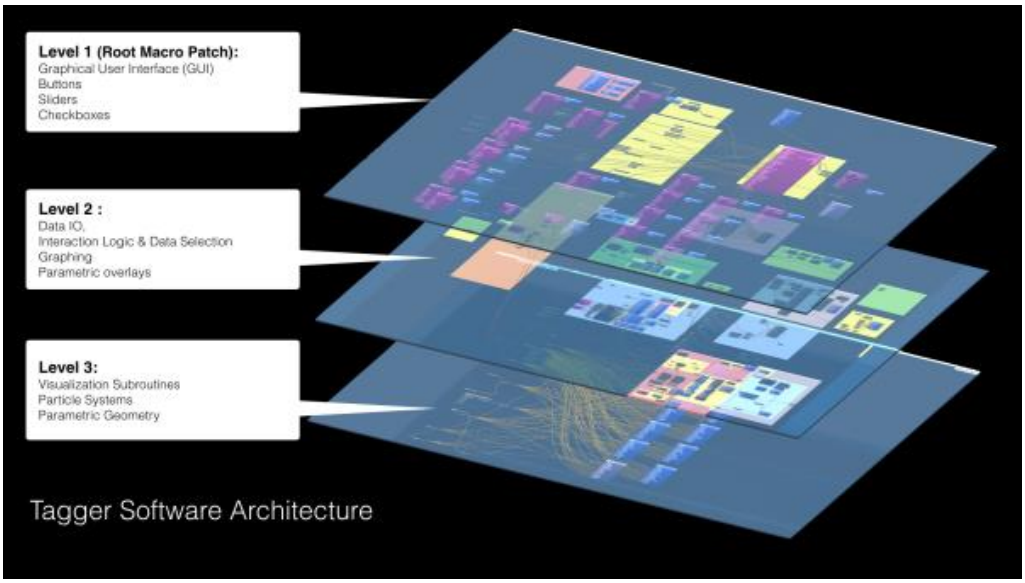


Figure 1: Tagger software architecture showing the relationship between user interface, data handling and visualization subroutines.

Tagger is a multi-layered Quartz Composer program constructed in a modular fashion (Fig. 1). Each layer consists of patches and macro-patches, which themselves contain complex subroutines for performing manipulations upon the data streaming into them and rendering to screen.

It has three main levels:

- Level 1 (Root Macro Patch): implements the majority of the GUI controls such as buttons, checkboxes and sliders) and draws the main GUI to screen.
- Level 2: implements the main functional modules for data IO, interaction logic and data selection, graphing and parametric overlays.

- Level 3: implements subroutines necessary for some higher level 2 modules to function such as particle systems and parametric geometry for visualization overlays.

See main text for overview of the Tagger GUI.

5 Supplement 5: Case Study Data

YEAR	Leap Year	SAMPLEs (min)	Header rows	Start Date	End Date	Rows	Records	% Uptime	File Size (MB)	Note
2000	Y	30	2	01/01/2000	31/12/2000	17285	17283	98.38	1.4	
2001	-	30	2	01/01/2001	31/12/2001	17165	17163	97.96	1.4	
2002	-	30	2	01/01/2002	31/12/2002	17236	17234	98.37	1.4	
2003	-	30	2	01/01/2003	31/12/2003	16793	16791	95.84	1.3	
2004	Y	30	2	01/01/2004	31/12/2004	13931	13929	79.29	1	Dates 1/6/04-28/7/04 missing
2005	-	30	2	01/01/2005	31/12/2005	16432	16430	93.78	1.2	
2006	-	30	2	01/01/2006	31/12/2006	17279	17277	98.61	1.4	
2007	-	30	2	01/01/2007	31/12/2007	16853	16851	96.18	1.3	
2008	Y	30	2	01/01/2008	31/12/2008	17438	17436	99.25	1.4	
2009	-	30	2	01/01/2009	31/12/2009	15092	15090	86.13	1.2	

Table 1: Case Study Data

6 Supplement 6: Video

A demonstration video may be seen here: https://vimeo.com/205325366

7 Supplement 7: Scalability

Tagger can interactively display data volumes ranging from very small kilobyte-size .csv samples to much larger megabyte samples. As indicated in Supplement 2, Tagger can be configured to interact with THREDDS servers, which enable interaction with terabyte or larger data sources that can be sliced (or subsampled) and streamed to Tagger for interactive reconnaissance analysis.

The Waverider Buoy case study involved 13MB of data and we have carried out further testing on larger file sizes to assess scalability. The dataset for testing larger file sizes comprises a time series from seismic station WRAB (one of the Warramunga Array stations, Northern Territory, Australia) close to the origin time of a large earthquake in Japan.

In the following tests (Table 1) we take Frame Rate as a proxy for overall compute performance of the display system. Computer is a 2013 Mac Pro Desktop with 1 x 3.7GHz Quad-Core Intel Xeon E5 CPUs, 32GB 1866 MHz DDR3 RAM, 2 x AMD FirePro D700 GPUS (total of 12Gb VRAM).

Tagger v0.6.1.3	Data Size (MB)	Total # Rows (no header)	Total Cols	Total Samples	Total Samples (3 Cols)	SS=1 Frame Rate	SS=10 Frame Rate	SS=100 Frame Rate
Test File (WRAB_1 .csv)	10.1	132120	9	1,189,080	396,360	19.7-20.0	19.1-20.0	19.2-20.0
Test File (WRAB_2 .csv)	20.1	264240	9	2,378,160	792,720	19.9-20.0	19.6-20.0	19.5-20.0
Test File (WRAB_3 .csv)	50.3	660600	9	5,945,400	1,981,800	19.3-20.0	19.5-20.0	19.4-20.0
Seismic (WRAB_4 .csv)	100.6	1321200	9	11,890,800	3,963,600	19.4-20.0	19.5-20.0	19.1-20.0

Table 1: Tagger Animation Performance (Frame Rate and File Size). Tagger parameters: SS = subsample; Max Queue Size = 1000; Y max value = 2500; Graphing is switched on, MetaData Display is on, 3 columns of data are plotted. 100 items are selected for tagging. Data were displayed for 10 minute periods of observation with no visual enhancement overlays.

Frame rate ranges stabilise at 19.9-20.0 frames per second (fps) regardless of the size of the source data. Lower frame rates are attributable to the number of data points drawn to screen and GPU performance, as well as background processes of the operating system. Tests indicate that the frame rate begins to drop when more than 10,000 data points are drawn to screen (10,000 points perform at 15fps; 20,000 points perform at 12fps) using the above configured computer system. Lower specification systems will exhibit reduced performance. The graphical display (GPU and graphics library performance) therefore contributes to the practical limit on scalability. Frame rates may also be affected temporarily by user interactions such as changing selection and display parameters. This is a normal consequence of GUI drawing processes in OpenGL and OS operations.

8 References

Abram, G 2000, “OpenDX Overview,” *Visualization Development Environments*, pp. 27–28.

Ahrens, JP, Woodring, J, DeMarle, DE, Patchett, J & Maltrud, M 2009, “Interactive remote large-scale data visualization via prioritized multi-resolution streaming,” in *Proceedings of the 2009 Workshop on Ultrascale Visualization*,pp. 1–10.

Apple Computer 2007, *Quartz Composer User Guide*,, accessed August 25, 2016, from <https://developer.apple.com/library/mac/documentation/GraphicsImaging/Conceptual/QuartzComposerUserGuide/qc_intro/qc_intro.html>.

Apple Computer 2016, “Xcode,”, accessed August 25, 2016, from <https://developer.apple.com/xcode/>.

Bostock, M, Ogievetsky, V & Heer, J 2011, “D^3: Data-Driven Documents.,” *IEEE transactions on visualization and computer graphics*, vol. 17, no. 12, pp. 2301–9.

Brown, D, Brownrigg, R, Haley, M & Huang, W 2012, “The NCAR Command Language (NCL)(version 6.0. 0),” *UCAR/NCAR Computational and Information Systems Laboratory, Boulder, CO.[Available online at http://dx. doi. org/10.5065/D6WD3XH5.]*.

Cinder Developers 2016, “Cinder C++ Library,”, accessed January 27, 2016, from <http://libcinder.org>.

Core-Plot Developers 2016, “Core- Plot,”, accessed August 25, 2016, from <https://github.com/core-plot/core-plot>.

Cowan, B & Kapralos, B 2014, “A Survey of Frameworks and Game Engines for Serious Game Development,” in *Advanced Learning Technologies (ICALT), 2014 IEEE 14th International Conference on*,pp. 662–664.

CSIRO 2016, “CSIRO Workspace,”, accessed August 25, 2016, from <https://research.csiro.au/workspace/>.

Cummings, R 1993, “System architectures using fibre channel,” in *Mass Storage Systems, 1993. Putting all that Data to Work. Proceedings., Twelfth IEEE Symposium on*,pp. 251–256.

Davis, E & Caron, J 2002, “5.7 THREDDS: Publishing, Cataloging, Describing, and Discovering Scientific Datasets Ethan R. Davis* John Caron Ben Domenico,”.

Demsar, J, Zupan, B, Leban, G & Curk, T 2004, *Orange: From experimental machine learning to interactive data mining*, Springer.

Emmott, S & Rison, S 2006, *Towards 2020 science*, Microsoft Research,.

Facebook Design 2016, “Origami Studio,”, accessed August 22, 2016, from <https://facebook.github.io/origami/>.

Folk, M, Heber, G, Koziol, Q, Pourmal, E & Robinson, D 2011, “An overview of the HDF5 technology suite and its applications,” in *Proceedings of the EDBT/ICDT 2011 Workshop on Array Databases*,pp. 36–47.

Fox, A, Griffith, R, Joseph, A, Katz, R, Konwinski, A, Lee, G, Patterson, D, Rabkin, A & Stoica, I 2009, “Above the clouds: A Berkeley view of cloud computing,” *Dept. Electrical Eng. and Comput. Sciences, University of California, Berkeley, Rep. UCB/EECS*, vol. 28, p. 13.

Friese, K-I, Herrlich, M & Wolter, F-E 2008, “Using Game Engines for Visualization in Scientific Applications,” in *New Frontiers for Entertainment Computing*, Springer, pp. 11–22.

Google Inc. 2016, “Google Charts,”, accessed January 23, 2016, from <https://developers.google.com/chart/>.

Gray, J, Liu, DT, Nieto-Santisteban, M, Szalay, A, DeWitt, DJ & Heber, G 2005, “Scientific data management in the coming decade,” , vol. 34, no. 4, pp. 34–41.

Gregg, C & Hazelwood, K 2011, “Where is the data? Why you cannot debate CPU vs. GPU performance without the answer,” in *Performance Analysis of Systems and Software (ISPASS), 2011 IEEE International Symposium on*,pp. 134–144.

Halic, T, Ahn, W & De, S 2015, “Optimization model for web based multimodal interactive simulations,” *Expert Systems with Applications*, vol. 42, no. 12, pp. 5245–5255.

Hansen, CD & Johnson, CR 2011, *Visualization handbook*, Academic Press.

Hey, AJ, Tansley, S, Tolle, KM & others 2009, *The fourth paradigm: data-intensive scientific discovery*, Microsoft Research Redmond, WA.

Hoetzlein, RC 2012, “Graphics performance in rich internet applications,” *IEEE computer graphics and applications*, vol. 32, no. 5, pp. 98–104.

Kessenich, J, Baldwin, D & Rost, R 2004, “The OpenGL shading language,” *Language version*, vol. 1.

Khronos Group 2015, “Vulkan,”, accessed January 27, 2016, from <https://www.khronos.org/vulkan/>.

Khronos Group 2016, “OpenGL ES 2.0 for the Web,”, accessed January 23, 2016, from <https://www.khronos.org/webgl>.

Kitware 2016, “Visualization Toolkit (VTK),”, accessed January 27, 2016, from <http://www.vtk.org>.

Kitware Inc. 2016, “Paraview,”, accessed August 25, 2016, from <http://www.paraview.org>.

Kosada Inc. 2016, “Kineme: Quartz Composer Stuff,”, accessed August 25, 2016, from <https://kineme.net>.

McClanahan, C 2010, “History and Evolution of GPU Architecture,” *A Paper Survey http://mcclanahoochie. com/blog/wpcontent/uploads/2011/03/gpu-hist-paper. pdf*.

Mell, P & Grance, T 2011, “The NIST definition of cloud computing,”.

Meloni, W 2015, “REV UP YOUR ENGINES.,” *Computer Graphics World*, vol. 38 , no. 1, pp. 18–23.

Morse, P, Reading, A, Lueg, C & Kenderdine, S 2015, “TaggerVR: Interactive Data Analytics for Geoscience-A Novel Interface for Interactive Visual Analytics of Large Geoscientific Datasets in Cloud Repositories,” in *Big Data Visual Analytics (BDVA), 2015*,pp. 1–2.

Myers, BA 1990, “Taxonomies of visual programming and program visualization,” *Journal of Visual Languages \& Computing*, vol. 1, no. 1, pp. 97–123.

National Research Infrastructure for Australia 2016, “NeCTAR Research Cloud,”, accessed January 27, 2016, from <https://www.nectar.org.au>.

Nativi, S, Caron, J & Domenico, B 2004, “NcML-G_ML: Encoding NetCDF Datasets Using GML.,” in *DEXA Workshops*,pp. 804–808.

Neider, J, Davis, T & Woo, M 1993, “OpenGL programming guide,”.

Pfister, GF 2001, “An introduction to the infiniband architecture,” *High Performance Mass Storage and Parallel I/O*, vol. 42, pp. 617–632.

Plotly Inc 2016, “Visualize Data, Together,”, accessed January 23, 2016, from <https://plot.ly>.

Processing Developers 2016, “Processing,”, accessed January 27, 2016, from <https://processing.org>.

Rew, R & Davis, G 1990, “NetCDF: an interface for scientific data access,” *Computer Graphics and Applications, IEEE*, vol. 10, no. 4, pp. 76–82.

Rijnieks, K 2013, *Cinder-Begin Creative Coding*, Packt Publishing Ltd.

Sanner, MF, Stoffler, D & Olson, AJ 2002, “ViPEr, a visual programming environment for Python,” in *Proceedings of the 10th International Python conference*,pp. 103–115.

Schroeder, WJ, Avila, LS & Hoffman, W 2000, “Visualizing with VTK: a tutorial,” *Computer Graphics and Applications, IEEE*, vol. 20, no. 5, pp. 20–27.

Sousa, TB 2012, “Dataflow Programming Concept, Languages and Applications,” in *Doctoral Symposium on Informatics Engineering*,.

Stehno, B 2012, *Rapid visualization development based on visual programming: developing a visualization prototyping language*, Proceedings of CESCG 2012: The 16th Central European Seminar on Computer Graphics.

Sutherland, WR 1966, *On-Line Graphical Specification of Computer Procedures.*,.

ThreeJS Developers 2016, “three.js,”, accessed January 23, 2016, from < http://threejs.org>.

Unidata 2016, “THREDDS Data Server,”, accessed January 27, 2016, from < http://www.unidata.ucar.edu/software/thredds/current/tds/>.

University of Ljubljana 2016, “Orange: Data Mining,”, accessed August 25, 2016, from <http://orange.biolab.si>.

Wernecke, J 1994, *The inventor mentor: programming object-oriented 3d graphics with open inventorTM, release 2*, Addison-Wesley.

Supplementary Material: Chapter 6

Supplement 1

Supplement 1 contains high-resolution figures (Figs.1-7), an overview of the application suite (Fig.8), UI design, video transcripts, information on underlying data values for our demonstration dataset, color and value distribution information for the presented case study, and supplementary figures (Figs.9 and 10) showing SMEAN2 (composite model) compared to a very recent model DETOX-P3 (single study).

S1.1 High Resolution Figures (Main Text Figs.1-7)

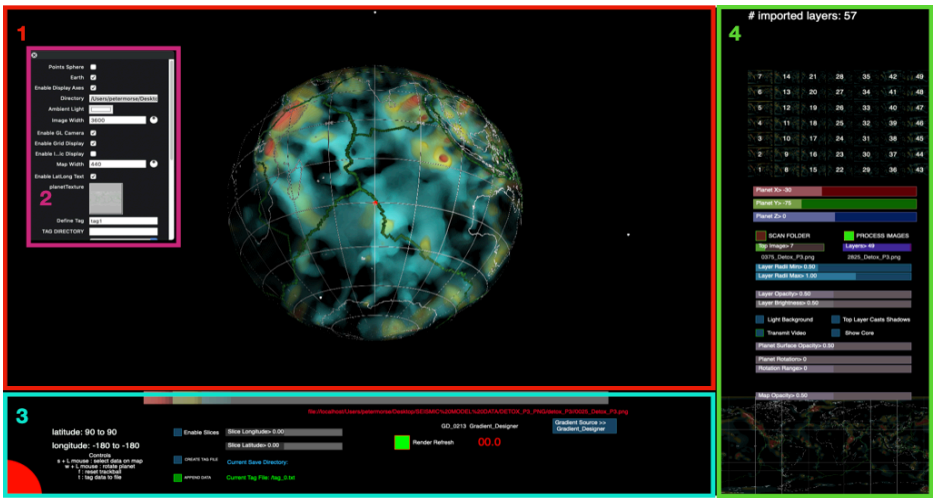


Fig. 1. PDT_V Application GUI (v.0.9.15). High resolution also available online.

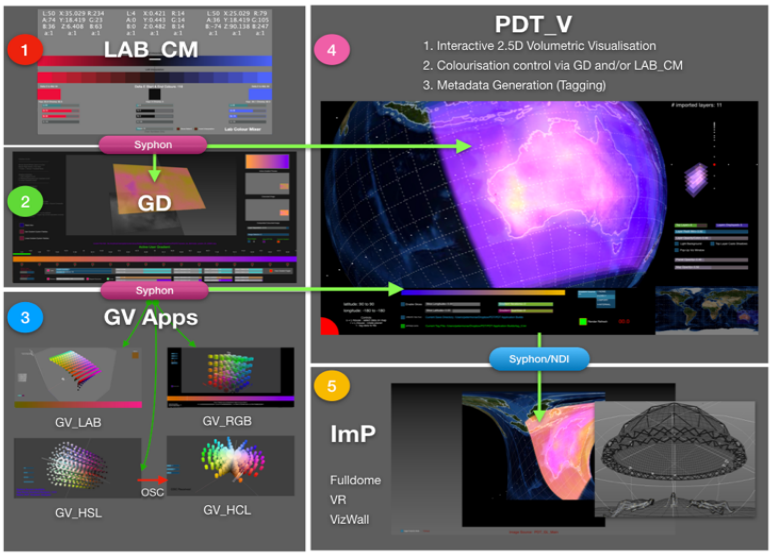


Fig. 2. Application suite workflow: 1) LAB Color Mixer (LAB_CM); 2) Gradient designer (GD); 3) Gradient Visualizer Apps (GV_LAB, GV_RGB, GV_HSL, GV_HCL); 4) Planetary Data Tagger Volumetric (PDT_V); 5) Immersive Player (ImP).

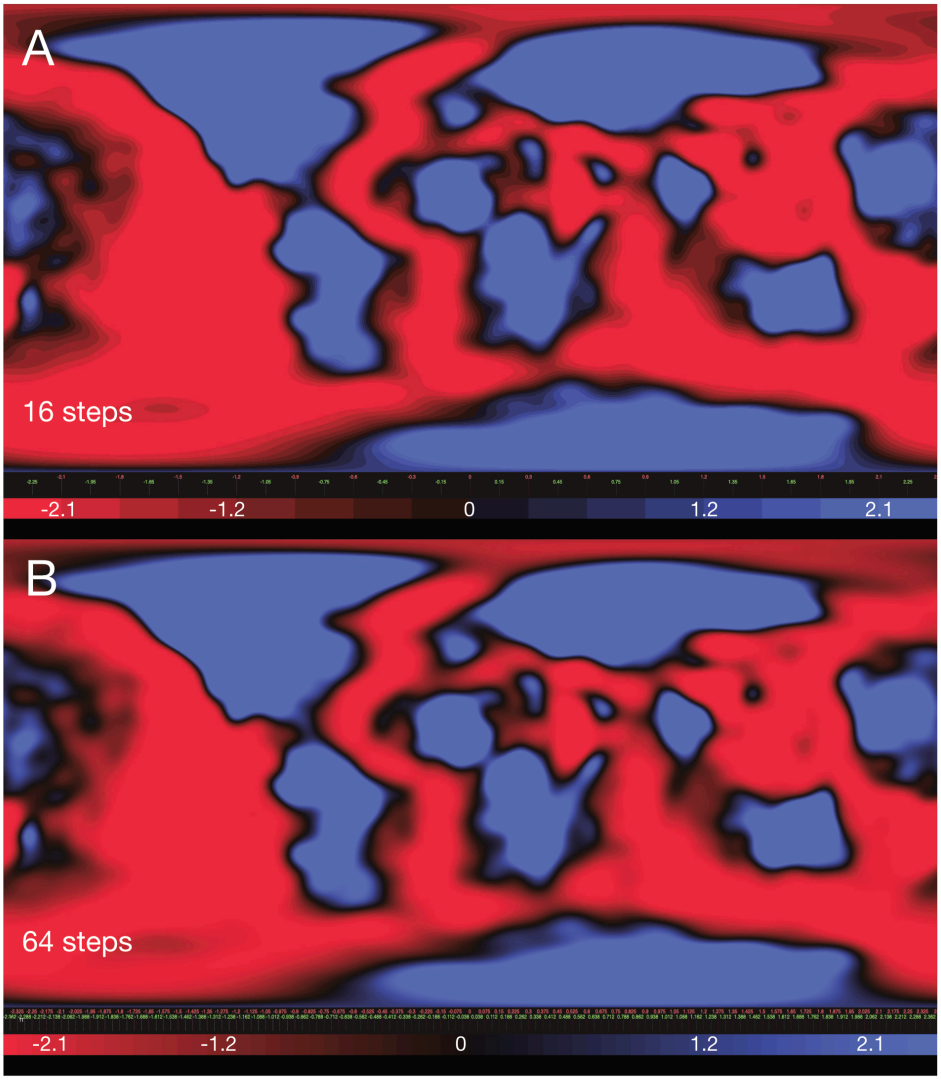


Fig. 3. Effect of gradient quantization upon SMEAN2 data visualization. A = 16 step gradient; B = 64 step gradient. For these single-layer images, value ranges are listed in Supplement 1,Table 2 and shown on the figure scale (+/- % variance from PREM).

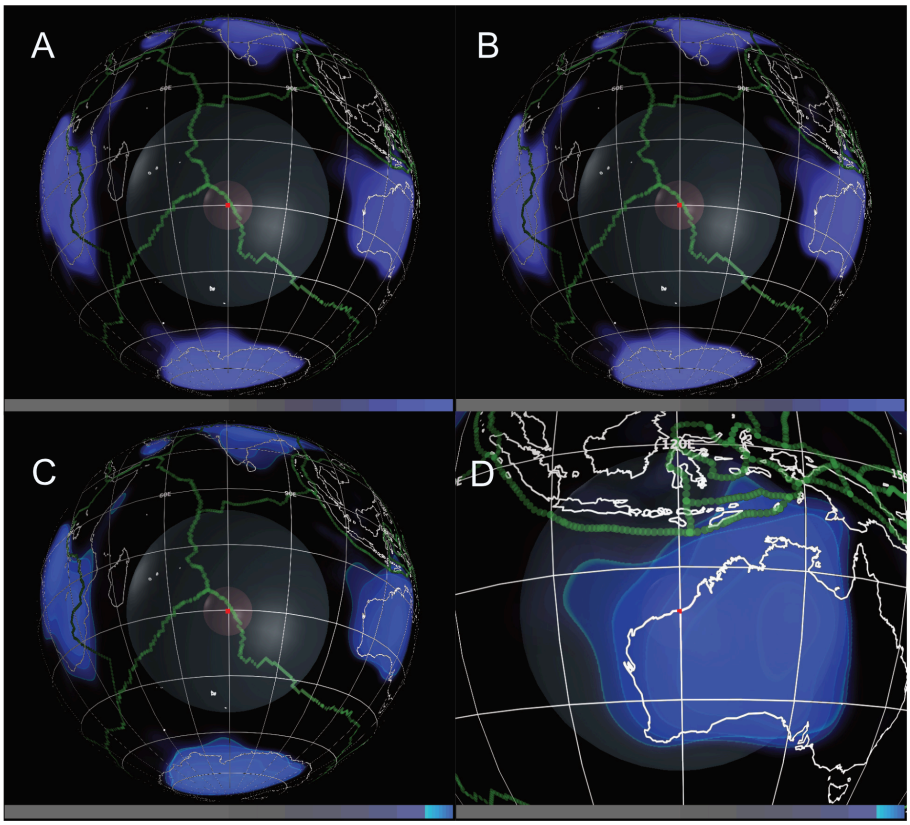


Fig. 4. Illustrative Visualization of the lithosphere/upper mantle in the Indian Ocean region and Australian continent. A and B employ a perceptually uniform colormap, whilst C and D employ a composited colormap. Value ranges are listed in Supplement 1, Table 2. Values are deliberately not shown on the figure itself because the visualizations display multiple composited layers.

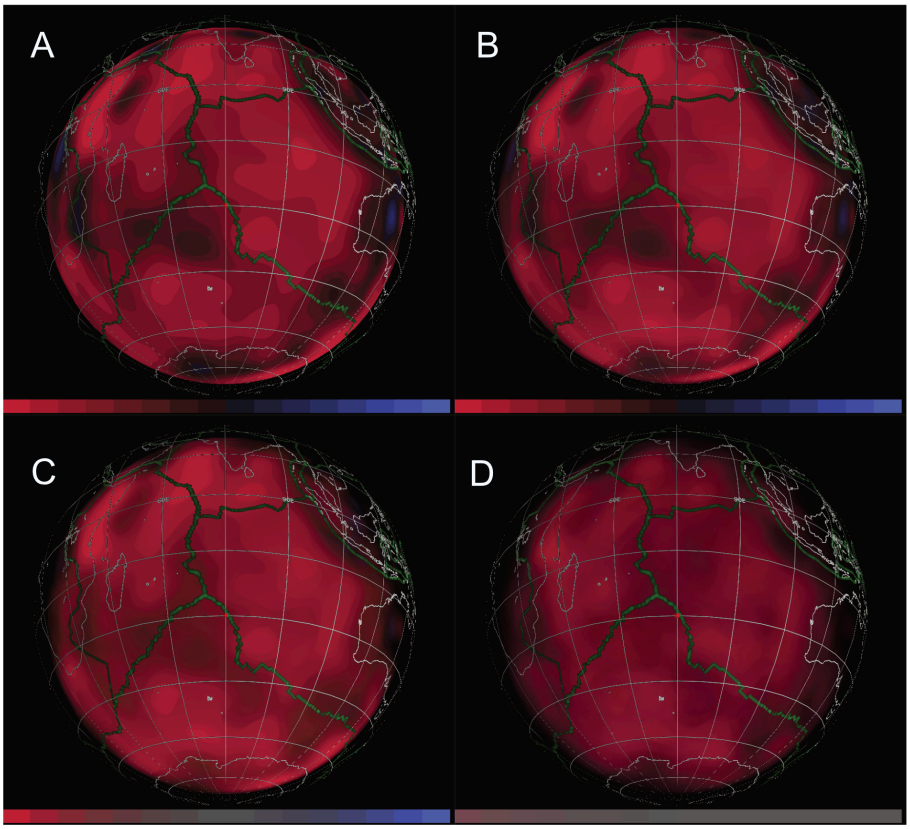


Fig. 5. Visualization of the deep mantle beneath the Indian Ocean region using a perceptually uniform colormap with systematically varying alpha values. See also note on value ranges / multiple layer compositing in the caption to Fig. 4.

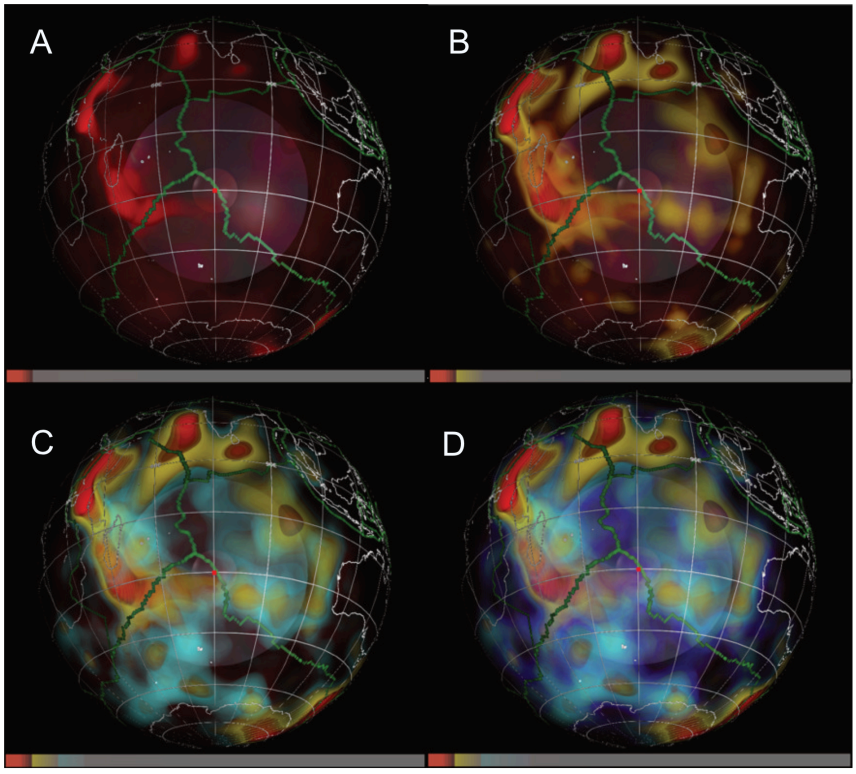


Fig. 6. Visualization of the deep mantle beneath the Indian Ocean region using systematically varying composite gradients. See also note on value ranges / multiple layer compositing in the caption to Fig. 4.

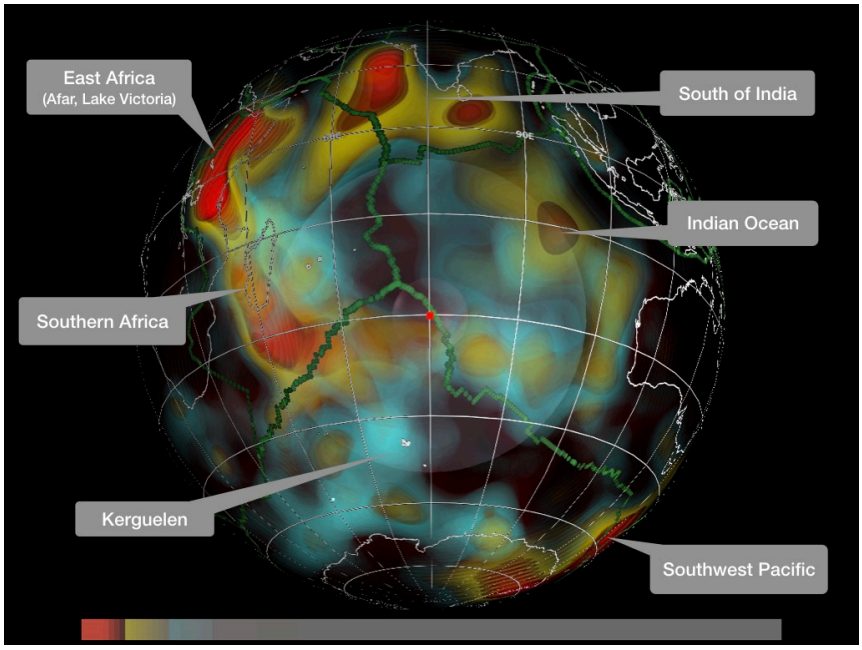


Fig. 7. Exploratory visualization of the deep mantle beneath the Indian Ocean region using the preferred composite gradient (as in Fig.6C, features of interest are identified). See also note on value ranges / multiple layer compositing in the caption to Fig. 4.

S1.2 Overview of Application Suite

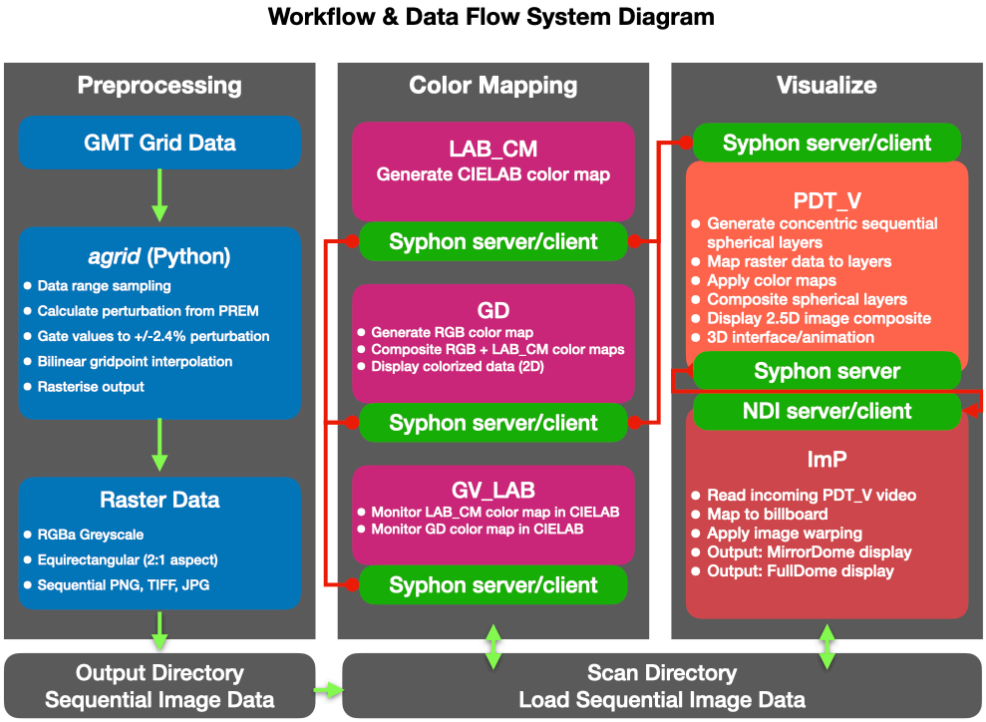


Fig. 8: Workflow and Data Flow System Design

Fig.8 provides a schematic representation of the visualization system workflow/algorithm and related data flow. It consists of three distinct steps:

- data preprocessing using *agrid* [1]
- color mapping using companion apps LAB_CM, GD and GV_LAB
- visualization using PDT_V and companion ImP apps

Step 1: Preprocessing.

The global seismic models SMEAN2 and DETOX-P3, are publicly available from the authors' repositories [2], [3]. SMEAN2 is provided in GMT grid format (*.grd), a variation of netCDF. DETOX-P3 is provided as space delimited text files (*.txt). We use the python package *agrid* [1] to format the perturbation data into a 3D-array with resolution 0.1 x 0.1 degrees (3600 x 1800 grid cells), and depths at 50km intervals from 25km to 2875km. Perturbation values are assigned to the grid cells by linear interpolation. Horizontally, we apply a much higher resolution than the provided data. *Agrid* natively saves data as netCDF together with

coordinates and metadata, but can also export raster formats as e.g. geoTIFF or Portable Network Graphics (PNG) format with defined resolution and extent. When exporting data in PNG format, the pixel value does not directly represent the perturbation ratio, but an 8-bit grayscale value (0-255). To map perturbation value to 8-bit RGB grayscale, a linear mapping function defines the value range. The data range, RGB-8 grayscale pixel values, and confirmation of stored value, are written to a text file.
https://github.com/TobbeTripitaka/agrid.

Step 2: Color Mapping

Output RGB greyscale PNG rasters are written to a local or network available directory that is accessed by selecting the data source directory via the pop-up parameter window of GD (Fig.2). Rasters are loaded into GPU VRAM, where color mapping is applied via LAB_CM and/or GD, with gradient image data shared via the Syphon client/server pipeline built into each application. Similarly, GV_LAB can monitor this Syphon image data to visualize gradients within CIELAB color space. Further details are provided in the main text.

Step 3: Visualize

PDT_V loads sequential raster data from the common image data directory, texture-mapping them to concentric spherical layers, with textures assigned by the sequential order of images. Spherical layers are generated in response to the number of detected images and displayed according to user interface selection sliders (Fig. 1). Recommended compositing operations and parameter settings are discussed in detail in the main text. A variety of display grids, alternative compositing functions, layer alpha interpolation modes, and OpenGL functions are also available through the parameters window. Further documentation detailing these and other UI functions is being developed online [4].

PDT_V visualizations are piped via a Syphon/NDI-compatible pipeline to companion ImP (Immersive Player) proof-of-concept applications under development in Vuo [5] and Unity [6]. Video data is mapped to billboards (OpenGL quads) in respective applications and reprojected using meshwarp transforms for MirrorDome [7] in Vuo, and a multi-camera approach for Fulldome format in Unity [8]. Whilst these are 2D projections, dome projection can maintain perceptually apparent 3D and stereopsis through interactivity and animation, when composited against a black background. A similar approach is trivial to implement for certain types of XR deployment, such as e.g. AR figures registered as physical book illustrations. Future work will implement a fully 3D pipeline within an appropriate development environment (see main text, 7.3 ImP, Future Work). Dome implementations are demonstrated in the accompanying video 1.

S1.3 UI Design

As noted QC was chosen for its graphical programming paradigm, live code performance and wide API support. This includes rapid UI development using conventional and bespoke UI elements, with the ability to easily modify the UI design and layout. QC encourages an MVC design pattern [9] and this is reflected in our current interface choices. An attraction in this approach is to functionally separate controllers from views, whereby, for instance, a controller UI could be implemented on a mobile device, with the view displayed on a dome or XR display that might notionally require quite different types of interface controls if they were to be displayed onscreen. We consider the rapid UI iteration approach conducive to effective exploration of this design space. UI design is both an art and a science, we anticipate future iterations of our software will refine design and utility with further user-testing and feedback.

S1.4 Video 1: PDT_V Data Visualization Suite Demonstration (1080P, Color, 4m 10s)

Overview of the 5 component applications and their purpose:

1. LAB_CM 'LAB Color Mixer': Creates colour gradients, including divergent gradients in CIELAB space
2. GD 'Gradient Designer': Creates and composites RGBA/CIELAB gradients
3. GV_LAB 'Gradient Visualizer LAB': Monitors gradient trajectories in CIELAB colour space
4. 4 PDT_V 'Planetary Data Tagger': Interactive visualization of global, subsurface data
5. ImP: Companion applications for collaborative displays (e.g. Dome)

This includes (times are seconds):

Application Suite

0:12 – Application Suite diagram
0:17 – 1] LAB_CM function
0:19 – 2] GD function
0:21 – 3] GV Apps functions
0:24 – 4] PDT_V function
0:28 – 5] ImP apps functions

PDT_V

0:36 – Basic operation (launch app, parameter Setting)
0:49 – [Underlying code (viewable, modifiable)]
1:05 – Configuration (loading data, viewing input images, selection of data range etc)
1:38 – Operation (interactive usage)
1:51 – Gradient import functions

PDT_V + Companion Apps
1:59 – PDT_V Color – interaction with application suite
2:05 – PDT_V, LAB_CM, GV_LAB interaction
2:23 – CIELAB Color pipeline via Syphon: LAB_CM to GD to PDT_V RGB colorspace
2:41 – GD RGBA gradient to PDT_V
3:03 – GD RGBA gradient peak isoluminance monitored in GV_LAB CIELAB colorspace

PDT_V metadata
3:19 – PDT_V demonstration of basic lat/lon metadata and tagging functions

PDT_V Dome demonstrations
3:35 – PDT_V output via ImP demo apps for collaborative Dome display systems
4:09 – End

The video is also available at the github repository: https://github.com/pemorse/data-visualization-tools

S1.5 Video 2: PDT_V Deep Earth Animations (1080P, Color, 4m 2m 55s)

0:07 – Fig. 4A animation (SMEAN2)
0:11 – Fig. 4C animation (SMEAN2)
0:18 – Fig.5D animation (SMEAN2)
0:27 – Fig.6A animation (SMEAN2)
0:43 – Fig.6B animation (SMEAN2)
0:59 – Fig.6C animation (SMEAN2)
1:15 – Fig.6D animation (SMEAN2)
1.21 – Figs.6A-D comparison animation (SMEAN2)
1.52 – DETOX-P3 (2020) PDT_V UI display animation
2:08 – DETOX-P3 (2020), SMEAN2 (2016) comparison animation

The video is also available at the github repository: https://github.com/pemorse/data-visualization-tools

S1.6 Data Availability

High-resolution images, animations, source data and software (PDT_V, GD, LAB_CM and companion apps) are available from: https://doi.org/10.5281/zenodo.3264036. Follow update link to current release or visit the github repository:

https://github.com/pemorse/data-visualization-tools

S1.7 **Table 1: SMEAN2 Data Values**

TABLE 1: SMEAN2 DATA VALUES AND RANGES
NOTE: PREM VALUES ROUNDED TO 6 DIGITS. FULL DATASET AVAILABLE AT [131].

LAYER	Depth (km)	Min (% from PREM)	Max (% from PREM)	Mean (% from PREM)	Median (% from PREM)	Standard Deviation	min normalised	max normalised	RGB8 Min	RGB8 Max
1	25.0	-12.537167	6.165030	-0.185687	-0.229789	2.594158	0.000000	1.000000	0.0	255.0
2	75.0	-7.394983	6.600928	0.269371	0.256496	2.621099	0.000000	1.000000	0.0	255.0
3	125.0	-6.562132	7.613202	0.444263	0.099111	2.840409	0.000000	1.000000	0.0	255.0
4	175.0	-4.738863	6.785055	0.353052	-0.176476	2.208107	0.000000	1.000000	0.0	255.0
5	225.0	-3.829831	4.778291	0.206040	-0.115778	1.460672	0.000000	1.000000	0.0	255.0
6	275.0	-3.145902	3.224936	0.109937	0.001807	1.006641	0.000000	1.000000	0.0	255.0
7	325.0	-2.899844	2.545706	0.038285	0.036319	0.749469	0.000000	1.000000	0.0	255.0
8	375.0	-2.754472	2.064139	-0.018044	-0.032036	0.647006	0.000000	0.929412	0.0	237.0
9	425.0	-2.598479	2.013088	-0.045849	-0.086228	0.614263	0.000000	0.917647	0.0	234.0
10	475.0	-2.387379	2.087699	-0.055653	-0.100785	0.586798	0.003922	0.933333	1.0	238.0
11	525.0	-2.201580	2.026648	-0.065911	-0.115830	0.563625	0.043137	0.921569	11.0	235.0
12	575.0	-2.057667	1.947126	-0.077949	-0.133492	0.547803	0.070588	0.905882	18.0	231.0
13	625.0	-1.837126	1.789225	-0.082672	-0.135929	0.514216	0.117647	0.874510	30.0	223.0
14	675.0	-1.550200	1.553244	-0.081709	-0.128233	0.483135	0.176471	0.823529	45.0	210.0
15	725.0	-1.410674	1.494265	-0.075232	-0.116560	0.447961	0.207843	0.811765	53.0	207.0
16	775.0	-1.322783	1.724645	-0.062500	-0.088478	0.413777	0.223529	0.858824	57.0	219.0
17	825.0	-1.288447	1.957141	-0.051326	-0.068609	0.393886	0.231373	0.905882	59.0	231.0
18	875.0	-1.245850	2.022829	-0.041926	-0.061203	0.374416	0.239216	0.921569	61.0	235.0
19	925.0	-1.209092	2.029760	-0.033433	-0.052292	0.369520	0.247059	0.921569	63.0	235.0
20	975.0	-1.166043	1.997701	-0.028334	-0.046763	0.359403	0.258824	0.917647	66.0	234.0
21	1025.0	-1.135847	2.000731	-0.022607	-0.044960	0.352944	0.262745	0.917647	67.0	234.0
22	1075.0	-1.143155	1.978502	-0.016629	-0.041201	0.350046	0.262745	0.913725	67.0	233.0
23	1125.0	-1.155052	1.880748	-0.015060	-0.042715	0.344082	0.258824	0.890196	66.0	227.0
24	1175.0	-1.295476	1.789571	-0.015331	-0.040631	0.340534	0.231373	0.874510	59.0	223.0
25	1225.0	-1.459068	1.739981	-0.015324	-0.037145	0.341497	0.196078	0.862745	50.0	220.0
26	1275.0	-1.509322	1.622419	-0.010708	-0.030393	0.336557	0.184314	0.839216	47.0	214.0
27	1325.0	-1.489953	1.457360	-0.004461	-0.014381	0.336833	0.188235	0.803922	48.0	205.0
28	1375.0	-1.453015	1.405202	0.002336	-0.000758	0.341604	0.196078	0.792157	50.0	202.0
29	1425.0	-1.500745	1.334362	0.007771	0.010614	0.344049	0.188235	0.776471	48.0	198.0
30	1475.0	-1.554559	1.262286	0.013361	0.015393	0.347449	0.176471	0.764706	45.0	195.0
31	1525.0	-1.607999	1.253379	0.018481	0.018903	0.352263	0.164706	0.760784	42.0	194.0
32	1575.0	-1.658969	1.271096	0.020850	0.021886	0.357647	0.152941	0.764706	39.0	195.0
33	1625.0	-1.732456	1.293703	0.021915	0.029071	0.369412	0.137255	0.768627	35.0	196.0
34	1675.0	-1.813557	1.326596	0.022642	0.030449	0.382630	0.121569	0.776471	31.0	198.0
35	1725.0	-1.891427	1.398548	0.025927	0.031986	0.392456	0.105882	0.792157	27.0	202.0
36	1775.0	-1.959613	1.466944	0.029406	0.036587	0.405744	0.090196	0.803922	23.0	205.0
37	1825.0	-2.009353	1.481889	0.033633	0.045913	0.417217	0.082353	0.807843	21.0	206.0

38	1875.0	-2.018736	1.435445	0.039782	0.066256	0.413776	0.078431	0.800000	20.0	204.0
39	1925.0	-2.026177	1.361617	0.046545	0.084898	0.413461	0.078431	0.784314	20.0	200.0
40	1975.0	-2.029571	1.307450	0.053848	0.098916	0.417775	0.078431	0.772549	20.0	197.0
41	2025.0	-2.012963	1.258869	0.061742	0.115972	0.423125	0.082353	0.760784	21.0	194.0
42	2075.0	-1.996606	1.299226	0.069426	0.128970	0.433585	0.082353	0.772549	21.0	197.0
43	2125.0	-2.001503	1.309167	0.076042	0.138586	0.447162	0.082353	0.772549	21.0	197.0
44	2175.0	-2.026090	1.259723	0.078473	0.140316	0.458341	0.078431	0.760784	20.0	194.0
45	2225.0	-2.072528	1.193548	0.080277	0.144964	0.473503	0.066667	0.749020	17.0	191.0
46	2275.0	-2.124321	1.274659	0.082383	0.149821	0.491540	0.058824	0.764706	15.0	195.0
47	2325.0	-2.174051	1.351282	0.083534	0.154740	0.503778	0.047059	0.780392	12.0	199.0
48	2375.0	-2.229149	1.435600	0.084843	0.163893	0.516665	0.035294	0.800000	9.0	204.0
49	2425.0	-2.278004	1.514336	0.088269	0.177229	0.532321	0.023529	0.815686	6.0	208.0
50	2475.0	-2.324041	1.540914	0.098854	0.202616	0.551421	0.015686	0.819608	4.0	209.0
51	2525.0	-2.391334	1.572682	0.111229	0.227755	0.579455	0.000000	0.827451	0.0	211.0
52	2575.0	-2.454632	1.632180	0.122759	0.256193	0.614765	0.000000	0.839216	0.0	214.0
53	2625.0	-2.548341	1.794859	0.127740	0.288420	0.655928	0.000000	0.874510	0.0	223.0
54	2675.0	-2.676420	1.962635	0.132281	0.315981	0.706745	0.000000	0.909804	0.0	232.0
55	2725.0	-2.805928	2.113784	0.137654	0.337643	0.766672	0.000000	0.941176	0.0	240.0
56	2775.0	-2.932368	2.260602	0.143835	0.347181	0.835109	0.000000	0.972549	0.0	248.0
57	2825.0	-3.057575	2.409748	0.150586	0.361787	0.910410	0.000000	1.000000	0.0	255.0
58	2875.0	-3.182209	2.565320	0.157908	0.374640	0.992048	0.000000	1.000000	0.0	255.0

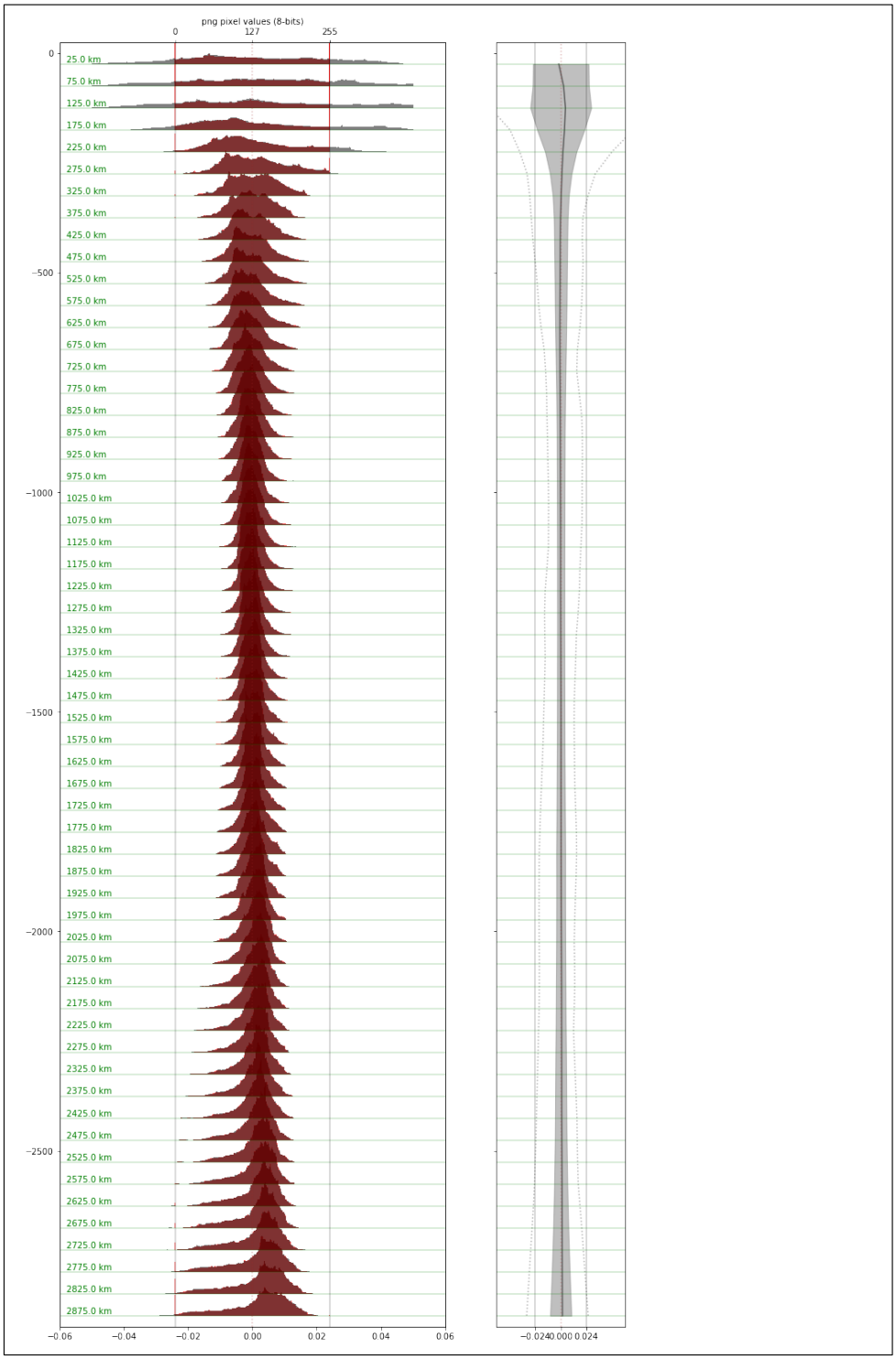
S1.8 **Table 2: Figure Color Values**

TABLE 2: FIGURE COLOR VALUES

Figure	STEPS/ LAYERS/ NORMALIZATION	% variance from PREM	LAB_CM (LAB+a values)	GD (HSLa Values)	PDT_V (alpha values)
3A	16 steps 1 layer (75km)	MIN: -7.394982814788820 MED: 0.25649648904800400 MAX: 6.60092830657959	R : 50, 74, 36 Bk : 4, 0, 0 B : 50, 36, -74 a: 1,1,1	GD LAB layer a=0.5	-
3B	64 Steps 1 layer (75km)	MIN: -7.394982814788820 MED: 0.25649648904800400 MAX: 6.60092830657959	R : 50, 74, 36 Bk : 4, 0, 0 B : 50, 36, -74 a: 1,1,1	GD LAB layer a=0.5	-
4A	16 Steps 6 Layers (75-325km) 0.99-0.95	See Table 1.	R : 50, 74, 36 Bk : 4, 0, 0 B : 50, 36, -74 a: 0,0,1	LAB_CM direct	a = 0.5
4B	16 Steps 57 Layers (75-2875km) 0.99-0.55	See Table 1.	R : 50, 74, 36 Bk : 4, 0, 0 B : 50, 36, -74 a: 0,0,1	LAB_CM direct	a = 0.5
4C	16 Steps 57 Layers (75-2875km) 0.99-0.55	See Table 1.	R : 50, 74, 36 Bk : 4, 0, 0 B : 50, 36, -74 a: 0,0,1	GD LAB layer a=0.5; **GD S15= Cyan-Blue** UC1: 0.57, 0.90, 0.50, 0.50; UC2:0.64, 0.90, 0.50, 0.50	a = 0.5
4D	16 Steps 57 Layers (75-2875km) 0.99-0.55	See Table 1.	R : 50, 74, 36 Bk : 4, 0, 0 B : 50, 36, -74 a: 0,0,1	GD LAB layer a=0.5; **GD S15= Cyan-Blue** UC1: 0.57, 0.90, 0.50, 0.50; UC2:0.64, 0.90, 0.50, 0.50	a = 0.5
5A	16 Steps 51 Layers (375-2875km) 0.94-0.55	See Table 1.	R : 50, 74, 36 Bk : 4, 0, 0 B : 50, 36, -74 a: 1,1,1	-	a = 1
5B	16 Steps 51 Layers (375-2875km) 0.94-0.55	See Table 1.	R : 50, 74, 36 Bk : 4, 0, 0 B : 50, 36, -74 a: 1,0,1	-	a = 0.5

5C	16 Steps 51 Layers (375-2875km) 0.94-0.55	See Table 1.	R : 50, 74, 36 Bk : 4, 0, 0 B : 50, 36, -74; a: 0.5,0,0.5	-	a = 0.5
5D	16 Steps 51 Layers (375-2875km) 0.94-0.55	See Table 1.	R : 50, 74, 36 Bk : 4, 0, 0 B : 50, 36, -74 a: 0.25,0,0	-	a = 0.5.
6A	16 Steps 51 Layers (375-2875km) 0.94-0.55	See Table 1.	R : 50, 74, 36 Bk : 4, 0, 0 B : 50, 36, -74 a: 0.1,0,0	LAB Layer a = 0.5; **GD S1: Red-Black** UC1 - RED H=0.0, S=0.9, L=0.5, A=0.5, UC2 – BLACK H=0.0, S0.9, L0.0, A0.5.	a = 0.5
6B	16 Steps 51 Layers (375-2875km) 0.94-0.55	See Table 1.	R : 50, 74, 36 Bk : 4, 0, 0 B : 50, 36, -74 a: 0.1,0,0	LAB Layer a = 0.5; **GD S1: Red-Black** UC1 - RED H=0.0, S=0.9, L=0.5, A=0.5, UC2 – BLACK H=0.0, S0.9, L0.0, A0.5; **GD S2: Yellow-Transparent** UC1 - YELLOW H0.15, S0.9, L0.21, A0.5; UC2 - TRANSPARENT H0.15, S0.9, L0.0, A0.0.	a = 0.5
6C	16 Steps 51 Layers (375-2875km) 0.94-0.55	See Table 1.	R : 50, 74, 36 Bk : 4, 0, 0 B : 50, 36, -74; a: 0.1,0,0	As 6B + **GD S3: Cyan-Transparent** UC1 - CYAN H=0.50, S=0.9, L=0.21, A=0.1, UC2 – TRANSPARENT H=0.50, S=0.9, L=0.0, A=0.0.	a = 0.5
6D	16 Steps 51 Layers (375-2875km) 0.94-0.55	See Table 1.	R : 50, 74, 36 Bk : 4, 0, 0 B : 50, 36, -74 a: 0.1,0,0	As 6C+ **GD S4: Blue-Transparent** UC1 - BLUE H=0.60, S=0.9, L=0.10, A=0.05, UC2 - H=0.50, S=0.9, L=0.0, A=0.0.	a = 0.5

S1.9 **SMEAN2 Histograms**



S1.10 SMEAN2 (2016), DETOX-P3 (2020) comparison (single study)

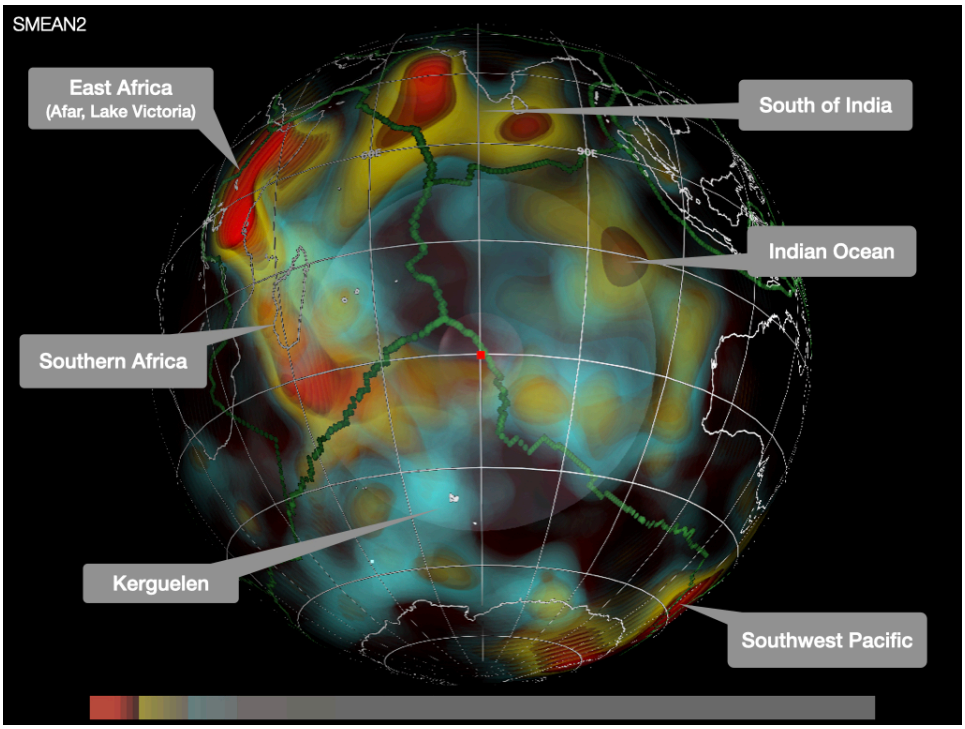


Fig.9: SMEAN 2 (2016)

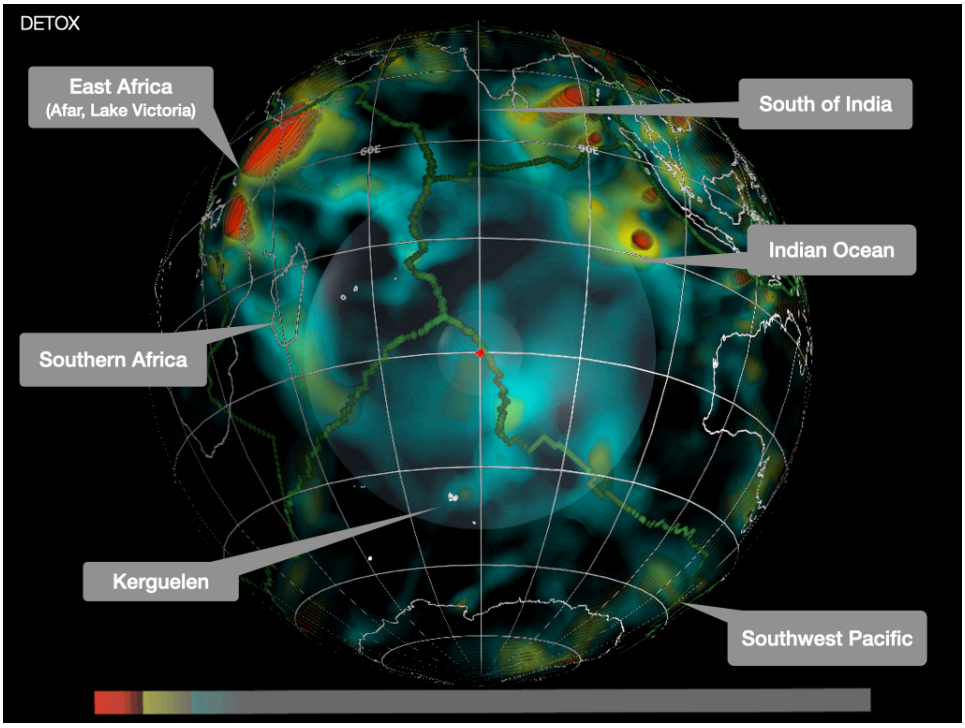


Fig.10: DETOX-P3 (2020)

Note on the figures: both Fig.9 and Fig.10 have had gradient settings applied as described in the main text.

References:

[1] T. Stål and A. M. Reading, "A Grid for Multidimensional and Multivariate Spatial Representation and Data Processing," Journal of Open Research Software, vol. 8, p. 2, Jan. 2020, doi: 10/ggvr5g.
[2] T. W. Becker, "SMEAN2 Composite Tomography Model," Chair in Geophysics: Seismic Tomography Repository, 2016. http://www-udc.ig.utexas.edu/external/becker/tdata.html#smean2 (accessed Jun. 02, 2020).
[3] K. Hosseini, "SubMachine: Web-based tools for exploring seismic tomography and other models of Earth's deep interior," SubMachine: Web-based tools for exploring seismic tomography and other models of Earth's deep interior, 2019. https://www.earth.ox.ac.uk/~smachine/cgi/index.php (accessed Jul. 04, 2019).
[4] P. E. Morse, "pemorse/data-visualization-tools: Gradient Designer Suite," Jan. 13, 2020. https://doi.org/10.5281/zenodo.3264036 (accessed Jan. 20, 2020).
[5] Kosada Inc., "Vuo," Vuo, 2019. https://vuo.org/ (accessed Jan. 21, 2020).
[6] Unity Technologies, "Unity," Unity, 2019. https://unity.com/frontpage (accessed Jan. 21, 2020).
[7] P. D. Bourke, "Fisheye warping for spherical mirror fulldome projection," Paul Bourke Personal Pages, Jul. 2012. http://paulbourke.net/dome/warpingfisheye/ (accessed Jan. 21, 2020).
[8] P. D. Bourke, "Creating fisheye views with the Unity3D engine," Paul Bourke Personal Pages, Aug. 2011. http://paulbourke.net/dome/unity3d/ (accessed Jul. 05, 2020).
[9] B. Shneiderman, C. Plaisant, M. Cohen, S. Jacobs, N. Elmqvist, and N. Diakopoulos, Designing the User Interface: Strategies for Effective Human-Computer Interaction, 6th ed. Pearson, 2016.

Supplementary Material: Technical Appendix

This appendix provides a synopsis of the software engineering approach employed in the software applications developed for the research. Discussion of accompanying data-management and networking procedures are detailed in the published papers (Chapters 4, 5 and 6) and in the supplementary material attending those publications, including extensive technical references.

- *Supplementary Material: Chapter 4* provides a detailed overview of QC (Section 3), and illustrates the structure of the Tagger software (Section 4). Application of the software is discussed in Chapter 4 (Morse et al., 2017), and demonstrated in the accompanying video.
- *Supplementary Material: Chapter 6* details the dataflow architecture of Gradient Designer (GD) and Planetary Data Tagger – Volumetric (PDT_V) applications and companion apps. Application of the software is discussed in Chapter 5 (Morse et al., 2019) and 6 (Morse et al., 2020), and documented in the accompanying videos.

As with any programming language, familiarity with the Quartz Composer (QC) integrated development environment (IDE) is required in order to understand the visual programming language (VPL) syntax and structure (Apple Inc, 2007).

The VPL structures are commented in the software hosted in the Github repository (Morse, 2020), providing documentation of the dataflow and visualization architecture. As noted in the main text, VPL architectures provide a strong degree of self-documentation by acting as kinds of 'wiring diagrams', analogous to electrical circuit diagrams. Thus many features of the software are self-explanatory when viewed in the QC IDE. Documentation for usage of the software is also available in the Github repository.

The software programs are multi-level and componentized, with self-contained sub-routines that perform specific functions such as providing UI elements, data IO, data and image processing routines and drawing OpenGL geometry to screen (Khronos Group, 2020). This enables components that perform specific functions to be cut and pasted between different versions of the software, facilitating rapid application iteration. Macropatches (similar to subroutines or objects in object-oriented programming) that fulfill specific functionality development are created from complex patch structures and stored in the QC patch library for re-use in companion programs that require similar functions (e.g. companion apps LAB_CM, GV_LAB). Examples of these include macropatches for:

- Parsing directories of files, determining number and type of files (e.g. binary or ASCII)
- Loading files into internal data structures (e.g. queues, dictionaries or structs)
- Mapping file data to geometries (e.g. graphing numerical values or assigning images to OpenGL textures)

- Performing logic operations upon data pipelines (e.g. Boolean tests for selections)
- Performing color space transformations (e.g. sRGB to LinearRGB to CIELAB)

The QC IDE identifies patches of varying types and macropatches by specific coloration and are commented, with their functionality described in the respective QC compositions. Likewise data pipelines are identified by intermediate 'splitter' or 'continuation' patches that indicate the type of data being passed and/or functions being deployed upon or using that data (Figure 1).

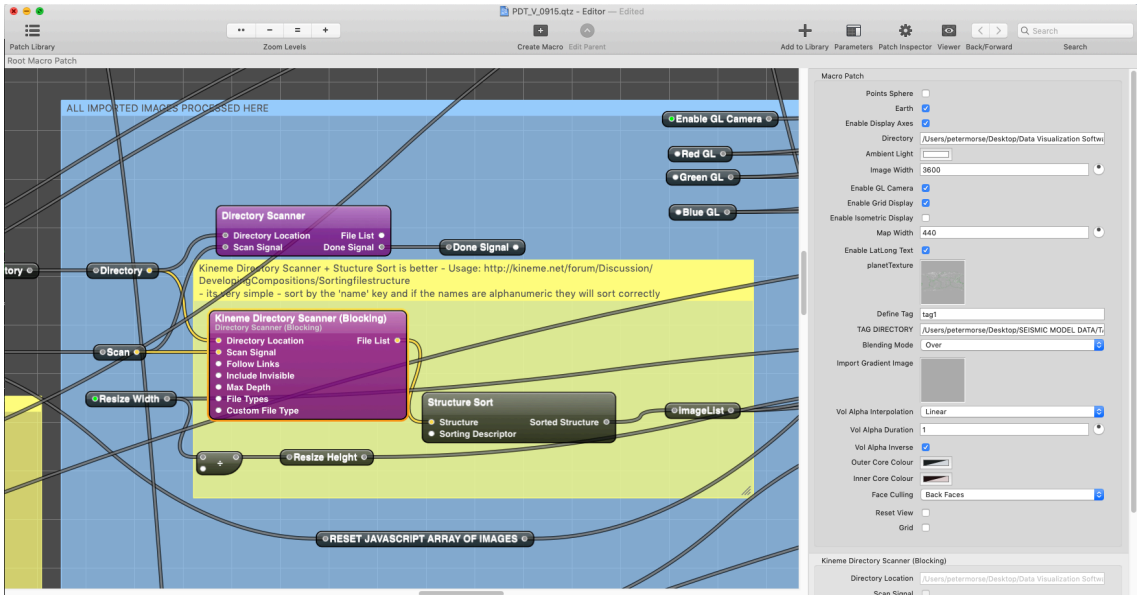


Figure 1: QC IDE with colored patches, comment note and macropatch parameter display

Traffic structures are organized for passing global or local variables between components or use transmission patches to send variables and data between encapsulated macro-patches.

The 'live-coding' capability of QC enables a robust programming and debugging methodology, facilitating iterative implementation of software engineering solutions. Accompanied by a 'lazy execution' model, the programmatic development approach enables the parallel development of computational solutions within a single project, enabling the programmer to retain 'solved' subroutines that can pass data to a tree of possible programmatic solutions downstream. These can be retained, copied and modified, deactivated or pruned as more optimal solutions are developed. In this respect the QC compositions contain a record of their own development, suggesting future development strategies.

Table 1 provides a high-level overview of the main software architecture developed for this research, representing the key compositional levels in each project, matched with the computational processing and logic operations that are undertaken.

Further documentation is available at Github:

https://github.com/pemorse/data-visualization-tools.

Software	Data IO	Process	Visualization	UI
Tagger	-File Reader Data -In: Macro Patch (Kineme Data Tools) 1D .txt, .csv numerical strings -Out: Write range selection values internal list to text file.	-Ingest numerical string data -Map value to QC x-coord -Map time to QC y-coord - Write Range selection values to internal list	-Instance Open GL UI -Draw Open GL line segments to window. -Draw Graph Scale to screen (Kineme Data Tools) -Draw Particle systems to Screen – parametric color driven by data value - Expose Parameters through parameter window	-Mouse -Trackball interface -Scroll Zoom -QC Widgets for UI (sliders, checkbox, lists) -Macro Patch for visual range selection and Data IO
GD	-File Reader Data -In: Macro Patch (Kineme Image Tools) 2D image files (.png, .jpg) -Out: File Writer 2D image files (.png, .jpg)	-Directory Scan -Enumerate files -Ingest image data -Process internal image structure on GPU -Apply color management (linear RGB, sRGB, custom Macro Patch CIELAB processing)	-Instance Open GL UI -Create OpenGL windows -Draw Open GL Quads to screen.	-Draw Open GL Quads to screen for UI gradient display -Draw GLSL custom shaders for gradient interpolation -Mouse -Trackball interface -Scroll Zoom -QC Widgets for UI (sliders, checkbox, lists) -Macro Patch for visual range selection and Data IO
PDT_V	-File Reader Data -In: Macro Patch (Kineme Image Tools) 2D image files (.png, .jpg) -Out: File Writer 2D image file sequence (.png, .jpg) or .mov	-Directory Scan -Enumerate files -Ingest image data -Process internal image structure on GPU -Apply color management (linear RGB, sRGB, custom Macro Patch CIELAB processing) -Texture map OpenGL sphere structure - Provide parametric compositing operations (Adjustable via UI)	-Instance Open GL UI -Create OpenGL windows -Iterate OpenGL sphere structure (Macro Patch) -Draw Open GL Sphere structure to screen. -Draw ancilliary views to screen (OpenGL quad structures) -Apply rotation (XYZ) -Apply rotation (t-interpolation)	-Draw Open GL Quads to screen for UI gradient display -Draw GLSL custom shaders for gradient interpolation -Draw OpenGL Sphere structure to Main Window -Mouse -Trackball interface -Scroll Zoom -QC Widgets for UI (sliders, checkbox, lists) -Macro Patch for visual range selection and Data IO

Table 1: Application Architecture

References:

Apple Inc, 2007. Quartz Composer User Guide [WWW Document]. URL https://developer.apple.com/library/mac/documentation/GraphicsImaging/Conceptual/QuartzComposerUserGuide/qc_intro/qc_intro.html (accessed 8.24.20).

Khronos Group, 2020. OpenGL Shading Language - OpenGL Wiki [WWW Document]. URL https://www.khronos.org/opengl/wiki/OpenGL_Shading_Language (accessed 8.22.20).

Morse, P., Reading, A., Lueg, C., 2017. Animated analysis of geoscientific datasets: An interactive graphical application. Computers & Geosciences 109, 87–94. https://doi.org/10/gcmxvw

Morse, P.E., 2020. pemorse/data-visualization-tools: Gradient Designer Suite [WWW Document]. https://doi.org/10.5281/zenodo.3264037

Morse, P.E., Reading, A.M., Stal, T., 2020. Exploratory volumetric deep Earth visualization by 2.5D interactive compositing. IEEE Trans. Visual. Comput. Graphics 1–1. https://doi.org/10/gh5n22

Morse, P.E., Reading, A.M., Stål, T., 2019. Well-Posed Geoscientific Visualization Through Interactive Color Mapping. Front. Earth Sci. 7, 274. https://doi.org/10/ggbjzq

END